

APS FOR z/OS

REFERENCE SUMMARY

Copyright © 2002 Micro Focus International Limited.
All rights reserved.

Micro Focus International Limited has made every effort to ensure that this book is correct and accurate, but reserves the right to make changes without notice at its sole discretion at any time. The software described in this document is supplied under a license and may be used or copied only in accordance with the terms of such license, and in particular any warranty of fitness of Micro Focus software products for any particular purpose is expressly excluded and in no event will Micro Focus be liable for any consequential loss.

Animator®, COBOL Workbench®, EnterpriseLink®, Mainframe Express®, Micro Focus®, Net Express®, REQL® and Revolve® are registered trademarks, and AAI™, Analyzer™, Application to Application Interface™, AddPack™, AppTrack™, AssetMiner™, CCI™, DataConnect™, Dialog System™, EuroSmart™, FixPack™, LEVEL II COBOL™, License Management Facility™, License Server™, Mainframe Access™, Mainframe Manager™, Micro Focus COBOL™, Object COBOL™, OpenESQL™, Personal COBOL™, Professional COBOL™, Server Express™, SmartFind™, SmartFind Plus™, SmartFix™, SourceConnect™, Toolbox™, WebSync™, and Xilerator™ are trademarks of Micro Focus International Limited. All other trademarks are the property of their respective owners.

No part of this publication, with the exception of the software product user documentation contained on a CD-ROM, may be copied, photocopied, reproduced, transmitted, transcribed, or reduced to any electronic medium or machine-readable form without prior written consent of Micro Focus International Limited.

Licensees may duplicate the software product user documentation contained on a CD-ROM, but only to the extent necessary to support the users authorized access to the software under the license agreement. Any reproduction of the documentation, regardless of whether the documentation is reproduced in whole or in part, must be accompanied by this copyright statement in its entirety, without modification.

U.S. GOVERNMENT RESTRICTED RIGHTS. It is acknowledged that the Software and the Documentation were developed at private expense, that no part is in the public domain, and that the Software and Documentation are Commercial Computer Software provided with RESTRICTED RIGHTS under Federal Acquisition Regulations and agency supplements to them. Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of The Rights in Technical Data and Computer Software clause at DFAR 252.227-7013 et. seq. or subparagraphs (c)(1) and (2) of the Commercial Computer Software Restricted Rights at FAR 52.227-19, as applicable. Contractor is Micro Focus, 9420 Key West Avenue, Rockville, Maryland 20850. Rights are reserved under copyright laws of the United States with respect to unpublished portions of the Software.

Table of Contents

1	About APS	7
	APS Painter Navigation	7
	Development Checklist	8
	Main Menu	12
	Painter Menu	13
	Help Facility	15
2	Application Painter	17
	Painter Navigation	17
	Application Painter	17
3	Control Files	23
4	Customization Facility	25
	Coding Macros	25
	Structures	26
5	Database Calls	31
	Coding Calls	31
	IDMS DB	32
	IMS DB	36
	SQL	41
	VSAM Batch	50
	VSAM Online	53
6	Data Communication Calls	57
	Coding Calls	57

- CICS 58
- IMS DC 65
- ISPF Dialog 70
- ISPF Prototyping 74
- 7 Data Element Facility..... 77**
 - Screens 77
 - Context Table Maintenance 77
 - Data Element Administration..... 77
 - Data Element Facility 77
 - Field/Screen Cross Reference 80
 - Screen Field Definition 80
 - Data Element Library Administration 80
- 8 Data Structures 83**
 - Painter Navigation..... 83
 - Coding Data Structures 83
 - Data Structures 84
- 9 ENDEVOR Interface 87**
 - Check In a Component 87
 - Check Out a Revision..... 89
 - Reports 90
- 10 Generation..... 93**
 - Generate Procedures..... 93
 - Parameters..... 94
 - Regenerate Procedures..... 102
- 11 Importers 105**
- 12 Keywords..... 109**
 - Placement in Code..... 109

Structures	113
Programming Commands	119
13 Limits	121
14 Online Express	125
Screen Navigation	125
SQL Database Access Screen Navigation	126
Screens	127
Control Points	158
15 Reports - APS-Generated	169
Report Descriptions	169
Report Production	175
16 Reports - Report Writer	179
Sample Structure	179
Requirements	180
Input-Output Section	180
File-Section	180
Report Section	180
Procedure Division	181
Mock-Ups	181
Structures	182
17 Reserved Words	189
18 S-COBOL	193
Coding Programs	193
Structures	195
19 Scenario Painter	201
Painter Navigation	201
Defining and Running a Scenario	202

20 Screen Painter205

Screen Navigation 205

Creating Screens 206

Screen Design Options 206

Field Attributes 207

Field Edits..... 208

Modifying Screens..... 219

Screen Generation Parameters..... 222

IMS Support Functions 226

ISPF Dialog Support Functions 228

21 User Help229

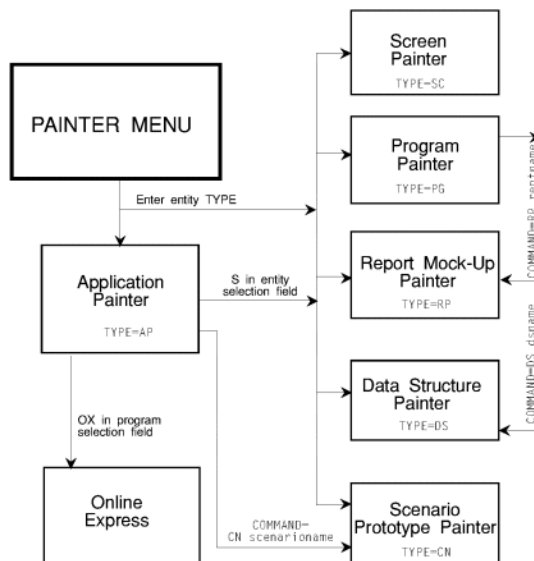
Creating 229

Revising Help Text 232

1 About APS

This chapter describes APS Painter navigation, includes a development checklist, and shows you APS menus and the Help Facility.

APS Painter Navigation



Development Checklist

Online Programs

Follow the steps below to develop your online applications. Refer to appropriate APS manual(s) for details about each step.

Step	Manual and Topic
1 Paint the application definition in the Application Painter.	APS User's Guide: <i>Paint the Application Definition</i> APS Reference: <i>Application Painter</i> <i>DB/DC Target Combinations</i> <i>See Application Painter</i> <i>See Limits</i>
2 Import the database definitions via the appropriate APS Importer.	APS User's Guide: <i>Import Database Definitions</i> APS Reference: <i>DDI Statements</i> <i>DDISYMB Flags</i> <i>See Importers</i>
3 Paint character screens in the Screen Painter.	APS User's Guide: <i>Paint Character Screens</i> <i>See Screen Painter</i> <i>See Limits</i>
4 Define screen field attributes.	APS User's Guide: <i>Paint Character Screens</i> APS Reference: <i>Attributes, Screen Fields</i> <i>See Screen Painter</i>
5 Create and run the application scenario in the Scenario Painter.	APS User's Guide: <i>Paint Character Screens</i> <i>See Scenario Painter</i>
6 Specify field edits for screen fields.	APS User's Guide: <i>Paint Character Screens</i>

Step	Manual and Topic
	APS Reference: <i>About Field Edits</i> <i>See Screen Painter</i>
7 Define the program functions in Online Express on the Program Functions, Alternate Functions, PF Key Functions, and Special Key Definition screens.	APS User's Guide: <i>Define Processing Logic</i> <i>See Online Express</i>
8 Specify character screen field mapping in Online Express on the Field Mapping screen.	APS User's Guide: <i>Define Processing Logic</i> <i>See Online Express</i>
9 Define database access in Online Express	APS User's Guide: <i>Define Database Access</i> <i>See Online Express</i> <i>See Limits</i>
10 Set environment and generation options from APS Generation Options.	APS User's Guide: <i>Generate the Application</i> APS Reference: <i>Bind and Translate Options, SQL</i> <i>Generator Options</i> <i>IDMS Options</i> <i>Job Control Cards</i> <i>MFS Function Keys</i> <i>MFS Trancode Construction</i> <i>MID MOD Reorder</i> <i>Panel Options, ISPF Dialog</i> <i>Precompiler Options</i> <i>See Generation</i>
11 Generate the application to the appropriate DB/DC targets from the Application Painter.	APS User's Guide: <i>Generate the Application</i> <i>See Generation</i>
12 Run and test the application.	APS User's Guide: <i>Generate the Application</i> <i>See Generation</i>
13 Produce reports that document the application in the Application Painter, Documentation Facility, or Report Generator.	APS Reference: <i>About Application Reports</i> <i>See Reports--APS-Generated</i>

Step	Manual and Topic
14 Optionally, check files into ENDEVOR.	APS User's Guide: <i>Using the APS/ENDEVOR Interface</i> APS Reference: <i>Checkin</i> <i>Checkout</i> <i>See ENDEVOR Interface</i>

Report Programs

Follow the steps below to develop your batch report programs. Refer to appropriate APS manual(s) for details about each step.

Step	Manual and Topic
1 Paint the application definition in the Application Painter.	See step 1 for Online Programs
2 Paint the report mock-up in the Report Mock-Up Painter.	APS User's Guide: <i>Create Reports with Report Writer</i> APS Reference: <i>Report Mock-Ups</i> <i>See Reports--Report Writer</i> <i>See Limits</i>
3 Specify the report logic in Program Painter using the APS Report Writer structures.	APS User's Guide: <i>Create Reports with Report Writer</i> APS Reference: <i>Report Writer Structures</i> <i>Specific keyword or structure</i> <i>See Reports--Report Writer</i> <i>See Limits</i>
4 Specify the program logic in the Program Painter, using keywords, and COBOL, COBOL/2, or S-COBOL structures, data structures, database calls, and APS/VSAM Batch calls.	APS User's Guide: <i>Create Reports with Report Writer</i> APS Reference: <i>Specific call, construct, keyword, or structure</i> <i>See Database Calls</i> <i>See Data Communication Calls</i>

Step	Manual and Topic
	<i>See Data Structures</i> <i>See Keywords</i>
5 Set environment and generation options from APS Generation Options.	See step 13 for Online Programs
6 Generate the application to the appropriate DB/DC targets from the Application Painter.	See step 14 for Online Programs
7 Run and test the application.	See step 15 for Online Programs
8 Produce reports that document the application in the Application Painter, Documentation Facility, or Report Generator.	See step 16 for Online Programs
9 Optionally, check files into the APS Version Control System.	See step 17 for Online Programs

Batch Programs

Follow the steps below to develop your batch applications. Refer to appropriate APS manual(s) for details about each step.

Step	Manual and Chapter
1 Paint the application definition in the Application Painter.	See step 1 for Online Programs
2 Specify the program logic in the Program Painter, using keywords, and COBOL, COBOL/2, or S-COBOL structures, data structures, database calls, and APS/VSAM Batch calls.	<i>APS User's Guide:</i> <i>Create Reports with Report Writer</i> <i>APS Reference:</i> Specific call, construct, keyword, or structure <i>See Database Calls</i> <i>See Data Communication Calls</i> <i>See Data Structures</i> <i>See Keywords</i>
3 Set environment and generation options from APS Generation Options.	See step 13 for Online Programs
4 Generate the application to the appropriate DB/DC targets from the Application Painter.	See step 14 for Online Programs
5 Run and test the application.	See step 15 for Online Programs

Step		Manual and Chapter
6	Produce reports that document the application in the Application Painter, Documentation Facility, or Report Generator.	See step 16 for Online Programs
7	Optionally, check files into the APS Version Control System.	See step 17 for Online Programs

Main Menu

Use the APS Main Menu to select any of the various APS screens or functions.

Option	Function
0	Transfer to APS Options Menu, a central facility for setting system-wide options.
1	Transfer to APS Painter Menu to select a specific painter.
2	Transfer to APS Dictionary Services to access: <ul style="list-style-type: none">• APS Importers to import IMS or IDMS data base specifications, DB2 catalog entities, VSAM files, or screens• Documentation Exporter to print reports• Data Element Facility to specify global definitions for APS screen fields• User Help Facility to create end-user help• Data Element Library Administration
3	Transfer to Prototype Execution to run the application prototype.
4	Transfer to APS Utilities Menu to access: <ul style="list-style-type: none">• APS Precompiler to compile and link non-painted source code• Custom utilities for system administration• DDI file utilities to unlock, backup, and restore DDI files.

Option	Function
	<ul style="list-style-type: none">SQL Subschema Maintenance Utilities to define, edit, browse, generate, and document XDB subschemas
5	Transfer to the ENDEVOR Interface Main Menu.
C	Display summary of new features in APS.
T	Transfer to APS Help Facility Tutorial.

Painter Menu

Use this screen to access any Painter, file, or ISPF data set. Then, you can create, modify, view, delete, or rename a component, print selected APS component reports, or create a new component identical to an existing one.

Field	Description and Values
Command	Leave Option and Member blank to display a list of entities. Enter one of the following values in selection field to the left of the component: <ul style="list-style-type: none">E Edit component. Create or modify a component. Default for all Painter Type selections except AP. Enter component name in Member field.D Delete component. Enter component name in Member field.B Browse component. Enter painter type in Type field and name of component in Member field. Default option for Type=AP.P Print APS report. Enter painter type in Type field and component name in Member field.C Create Like. Enter painter type in Type field, name of existing component in Member field, and name of new component in Newname field.

Field	Description and Values			
	R	Rename component. Enter painter type in Type field, name of existing component in Member field and new name for component in Newname field.		
Type	Painter type:			
	AP	Application Painter		
	SC	Screen		
	PG	Program		
	CN	Scenario		
	RP	Report		
	DS	Data Structure		
Member	Component name.			
	Programs:	First character alphabetic; remaining characters alphabetic, numeric, @, \$, or #; 8-character maximum.		
	Screens:	First character alphabetic; remaining characters alphabetic or numeric; 7-character maximum for CICS or ISPF prototyping, 8-character maximum for IMS or ISPF Dialog.		
	Data structures:	First character alphabetic; remaining characters alphabetic or numeric; 8-character maximum.		
	Report mock-ups:	First character alphabetic, @, or #; remaining characters alphabetic, numeric, @, \$, or #; 8-character maximum.		
Screen Size	Size of application screens.			
	Value	Screen Size	Fits on Terminal Size	
	MOD2	24 x 80	MOD2 3 4 5	
	MOD3	32 x 80	MOD 3 4	
	MOD4	43 x 80	MOD4	
	MOD5	27 x 132	MOD5	

Help Facility

Access online help from the APS Main Menu, or by entering **H** or pressing **F1** from any APS screen. Use any of the following commands to navigate the facility:

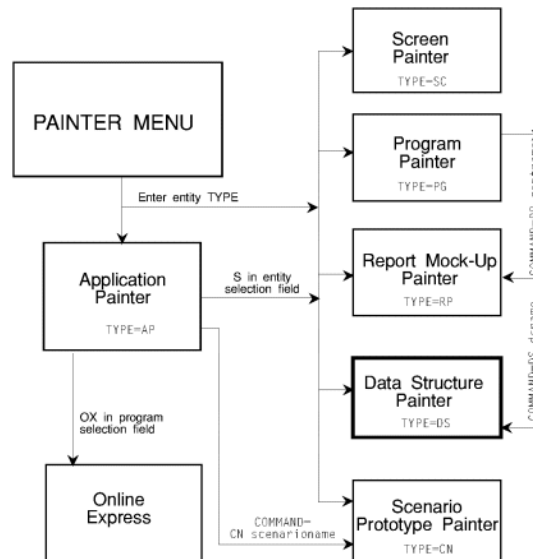
Tutorial Commands

B BACK	Return to previous screen
END F3 F15	Return to invoking screen
ENTER	Display next screen
SKIP	Advance to next Help topic
T TOP	Display Main Menu
U UP	Display a higher-level list of topics

2 Application Painter

This chapter describes the navigation and use of the APS Application Painter.

Painter Navigation



Application Painter

Application system entities are identified and entity relationships are defined in the Application Painter. The Application Painter screen also serves as a customized menu to transfer to another painter; print

definition reports; generate, compile, and link programs, screens, or the entire application.

Field	Description and Values		
mode field	If EDIT or blank, screen available for editing; if BROWSE (default), screen available only as a menu.		
prompt field	Displays short messages in upper right corner.		
Author	Name to appear in AUTHOR statement of generated program and as identifier on Application Definition Report.		
Screen Size	Size of application screens.		
	Value	Screen Size	Fits on Terminal Size
	MOD2	24 x 80	MOD2 3 4 5
	MOD3	32 x 80	MOD 3 4
	MOD4	43 x 80	MOD4
	MOD5	27 x 132	MOD5
DC Target	Target data communications environment:		
	ISPF (or TSO)		For ISPF (default)
	IMS		For IMS
	CICS (or CIC)		For CICS
	DDS		For OS400
	DLG		For ISPF Dialog
DB Target	Target database environment:		
	VSAM (or VSM)		For VSAM (default).
	DLI (or IMS)		For IMS DB
	VSAM		For SQL
	IDMS		For IDMS DB
selection fields	Any of the following:		
	B	Transfer to Browse mode to browse entity.	
	BD	BIND DB2 program.	
	E	Transfer to Edit mode to create or modify entity.	
	G	Generate, compile, and link entity.	
	OX	Create or modify Online Express program.	

Field	Description and Values
	R Print definition report.
	S Access entity.
Programs	1-8 char program name; 1st char alphabetic; remaining chars alphabetic, numeric, @, \$, or #.
Screens	1-8 char screen or screen name for IMS/MFS, PM, or ISPF Dialog, 1-7 char for ISPF prototyping and CICS; 1st char alphabetic; remaining chars alphabetic or numeric. Enter *BATCH for batch program, *STUB for program stub.
<hr/>	
Note: If global screen, enter * in adjacent Program field.	
<hr/>	
IO	Screen type: I Input only O Output only IO Input and output
Reports	1-8 char report mock-up name; 1st char alphabetic @, or #; remaining chars alphabetic, numeric, @, \$, or #.
Data Str	1-8 char data structure name; 1st char alphabetic; remaining chars alphabetic or numeric.
Type	Data structure type: WS Working-Storage Section (default) LK Linkage Section CA APS Commarea
SBSC/PSB	Name of subschema or PSB.
USERMACS	USERMACS member name to be INCLUDED in program. Loc Location of MAC MBR INCLUDE statement: T Top of source file (default) IO Top of Input-Output Section FD Top of File Section LT Top of Linkage Section

Field	Description and Values
LK	In Linkage Section after data structures or Customization Facility symbols/user macro calls
LB	Bottom of Linkage Section
RP	Top of Report Section
WT	Top of Working-Storage Section
WS	In Working-Storage Section after data structures, Customization Facility symbols/user macro calls, or SCRNLIST
WB	Bottom of Working-Storage
CA	Top of APS Commarea
B	Bottom of source file

Commands

Enter any of the following commands in the Command field.

Command	Description
<i>BIND [ALL]</i>	Submit a DB2 bind for all application programs found in DBRMLIB.
<i>BOPT</i>	Transfer to Bind Options.
<i>CN SCEN scenarioname</i>	Transfer to Scenario Prototype Painter; display scenario.
<i>DS datastructurename</i>	Transfer to Data Structure Painter; display data structure.
<i>GENERATE GEN type name</i>	Generate, compile, and link. <i>Type=</i> <i>MS</i> Multiple map mapset <i>PG</i> Programs <i>SC</i> Screens blank Entire application <i>Name= entityname, mapsetname, or ALL;</i> not applicable when <i>type</i> = blank.
<i>HELP F1 F13</i>	Display Help Tutorial.

Command	Description
<i>OPT [number]</i>	Transfer to APS Options Menu. If specify number, transfer to submenu, as follows: <ol style="list-style-type: none"> 1 Generator Options 2 Project Group Environment 3 Precompiler Options 4 Report Options 5 Bind Options 6 Job Control Cards 7 IDMS DB Options
<i>PG programname</i>	Transfer to Program Painter; display program.
<i>REPORT type name</i>	Print Definition reports. <i>Type=</i> <ul style="list-style-type: none"> <i>CN</i> Scenarios <i>DS</i> Data structures <i>PG</i> Programs <i>RP</i> Report mock-ups <i>SC</i> Screens blank Entire application <i>Name= entityname, mapsetname, or ALL;</i> not applicable when <i>type = CN</i> or blank.
<i>ROPT</i>	Transfer to Report Generator Options.
<i>RP reportname</i>	Transfer to Report Mock-Up Painter; display mock-up.
<i>RUN scenarioname</i>	Execute named scenario.
<i>SC screenname</i>	Transfer to Screen Painter; display screen.

3 Control Files

Use APS control file variables to control certain functions. Each control file contains documentation on its variables. Look in the APS CNTL directory or library for these files:

APS CNTL File	Environment or Function Controlled
APCICSIN	CICS
APSDBDC	Database and data communication calls
APFEIN	Field edits
APDLGIN	ISPF Dialog
APIMSIN	IMS DB and DC
APDB2IN	SQL
APVSAMIN	VSAM
APHLPIN	User Help Facility database

4 Customization Facility

This chapter describes coding macros and the structures used by the Customization Facility.

Coding Macros

Observe the following when coding macros:

Argument Lists	Arguments must be separated by a comma. Spacing between arguments is optional. Arguments are positional; to omit, code a comma in its place (except if it is the last argument). Enclose the argument list in parentheses.
ASCII Characters	All ASCII characters are valid except special characters such as tab and line feed.
Comments	Code <code>%* comment</code> .
Continuation	Code percent symbol, space, ellipsis, space (<code>% ...</code>).
End of a Macro	The function of a Customization Facility structure remains in effect until any statement occurs at the same or less indentation as the first line of the structure.
Indentation	Be consistent with the number of spaces that you indent; we recommend four spaces.
Literal Strings	Use single (<code>'</code>) or double (<code>"</code>) quotation marks to delimit literal strings; begin and end the string with the same character. You may use both single and double quote delimiter pairs in the same macro; when using the single quote delimiter, and you want to use an apostrophe within the string, use two single quotes.
Source Code	statement blocks may consist of Customization Facility, COBOL, and S-COBOL code.

Structures

BEGIN

```
% BEGIN
```

Built-In Functions

```
EPILOGUE function&APS-EPILOGUE
&APS-FULL
&APS-HALF
&APS-INDENT
&APS-PROGRAM-ID
&APS-PSB-NAME
&columnnumber+source
&COMPILETIME
&DEFINED( $macroname|&variablename)
&DEFVAL
&INDEX( 'whole string', 'character string')
&LENGTH( string)
&NUMERIC( &variablename)
               "literal string"
&PARSE( &string[, &variable
               number
               ])
&SUBSTR( string, startcolumn[, length])
&dataname+suffix
```

DECLARE

```
% DECLARE &fieldname(&sub1)[...(&subN)]
    [ % &declarepart1          [Xn|Nn|REDEFINES]
    .
    .
    % &declarepartN          [Xn|Nn|REDEFINES]]
```

DEFINE

```
% DEFINE $macroname [&formalarg1[, &formalarg2,  
% ... &formalarg3, ..., &formalarg1000]]  
    statementblock  
[% END]
```

END

```
[% END]
```

Error Handling

```
% SET FATAL messagetext  
% SET ERROR messagetext  
% SET WARNING messagetext  
% SET INFO messagetext  
% SET TRACE ERROR [NOINFO]  
% SET NOTRACE  
% SET WRITE-CONTROL
```

ESCAPE

```
% ESCAPE
```

IF/ELSE-IF/ELSE

```
% IF logicalterm1  
    statementblock  
% ELSE-IF logicalterm2  
    statementblock  
.  
.  
% ELSE-IF logicaltermN  
    statementblock  
% ELSE  
    statementblock  
[% END]
```

INCLUDE

```
% INCLUDE ddname[(membername)] [submember]
```

LOOKUP

```
% LOOKUP &declarepart operator searchvalue
                                     [FROM value [valueN ...]]
    statementblock1
[% ELSE-IF logicalterm
    statementblock2]
    .
    .
[% ELSE
    statementblockN]
[% END]
```

Macro Invocation

```
$macroname [(actualarg1[, actualarg2,
% ... actualarg3, ..., actualarg1000])]
```

REPEAT

- Format 1:

```
% REPEAT
    statementblock
% UNTIL|WHILE logicalterm
    [statementblock]
[% END]
```

- Format 2:

```
% REPEAT VARYING|R-V &variable
% ... [FROM &variable|literal|arithmeticexpression]
% ... [BY &variable|literal|arithmeticexpression]
    [statementblock]
% ... UNTIL|WHILE logicalterm
    [statementblock]
[% END]
```

- **Format 3:**

```
% REPEAT VARYING|R-V &variable
% ... [FROM &variable|literal|arithmeticexpr]
% ... [BY &variable|literal|arithmeticexpr]
% ... THRU|TO &variable|literal|arithmeticexpression
[% ... OR THRU|TO &variable|literal|arithmeticexpression]
[% ... OR THRU|TO &variable|literal|arithmeticexpression]
[% ... OR THRU|TO &variable|literal|arithmeticexpression]
    [statementblock]
[% END]
```

SET statements

Relocation statements:

```
% SET COMMUNICATION [SECTION]
% SET DATA [DIVISION]
% SET FILE-CONTROL
% SET FILE [SECTION]
% SET LINKAGE [SECTION]
% SET PROCEDURE
% SET SPECIAL-NAMES
% SET WORKING-STORAGE
% SET END-WORKING-STORAGE
```

Control statements:

```
% SET AUXILIARY-OUTPUT
% SET NORMAL-OUTPUT
% SET BLANK
% SET NOBLANK
% SET CONVERT-LOWER-CASE
% SET PRESERVE-LOWER-CASE
% SET DELIMITERS-OPTIONAL
% SET DELIMITERS-REQUIRED
% SET EPILOGUE [$macroname]
% SET EVAL-BRACKETS 'leftright'
[% SET EVAL-BRACKETS 'leftright']
% SET LEFT-MARGIN column
% SET RIGHT-MARGIN column
% SET ERROR messagetext
% SET FATAL messagetext
% SET INFO messagetext
% SET WARNING messagetext
% SET LOOP-LIMIT
SET TRACE ERROR [NOINFO]
```

```
SET NOTRACE
% SET WRITE-CONTROL[-LIMIT number]
.
.
[% SET NOWRITE-CONTROL]
```

UNTIL/WHILE

```
% UNTIL|WHILE logicalterm
    statementblock
[% END]
```

&variable

```
% &variable = stringterm|numericterm
```

5 Database Calls

This chapter describes APS database calls.

Coding Calls

Observe the following when coding database calls:

- Code the command using COBOL indentation.
- Arguments can be one of the following:
 - Variable
 - 1- to 7-digit number
 - Literal string delimited by single or double quotes
- Separate each command component with a space, unless indicated otherwise in the syntax.
- Some arguments are positional. Code positional arguments in the order shown in the syntax for each command.
- To omit a positional argument, code an asterisk (*) in its place (unless it is the last argument).
- A command can be continued on as many as 101 subsequent lines by coding an ellipsis followed by a space. Break a command for continuation at any blank space, but do not break a parameter.
- Comments:
 - Code anywhere in your program:

```
-KYWD-      12-*-----20---*--
/*          commentline
```

- Code in Procedure Division only:

```
-KYWD- 12-*----20---*----
/*commentline
```

- Continuation is an ellipsis followed by a space (...) and must continue on the immediate subsequent line.

IDMS DB

DB-BIND

- Format 1:

```
DB-BIND REC [recordname]
```

- Format 2:

```
DB-BIND RUN-UNIT
... SUBSCHEMA name|NODENAME name|DBNAME name
```

DB-CLOSE

```
DB-CLOSE ALL
```

DB-ERASE

```
DB-ERASE REC recordname [PERM|SELECT|ALL]
```

DB-GET

```
DB-GET REC [recordname]
```

DB-MODIFY

DB-MODIFY REC recordname

DB-OBTAIN

- Format 1, based on a CALC key or an indexed or sorted set:

```
DB-OBTAIN REC|REF recordname WHERE keyname = value [NEXT]
... [HOLD] [EXCLUSIVE]
```

- Format 2, where the most recently retrieved occurrence of a record type is current of run unit:

```
DB-OBTAIN REC|REF recordname CURRENT [HOLD] [EXCLUSIVE]
```

- Format 3, where the most recently retrieved occurrence of any record in the set is current of run unit:

```
DB-OBTAIN REC|REF IDMSREC SET setname| AREA areaname CURRENT
... [HOLD] [EXCLUSIVE]
```

- Format 4, based on database address:

```
DB-OBTAIN REC|REF IDMSREC|recordname
... WHERE DBKEY = value [HOLD] [EXCLUSIVE]
```

- Format 5, where the set owner is obtained when the record is unknown:

```
DB-OBTAIN REC|REF IDMSREC
... SET setname OWNER [HOLD] [EXCLUSIVE]
```

- Format 6, based on a CALC key or an indexed or sorted set, using a valid operator:

```
DB-OBTAIN [REF recordname1] REC|REF recordname2
... WHERE sortkey operator value
... [SET setname [RESET]]|[RESET] [HOLD] [EXCLUSIVE]
```

- Format 7, based on position within set:

```
DB-OBTAIN [REF recordname1] REC|REF recordname2
... [SET setname]
... [WHERE SEQUENCE = number|FIRST|LAST|PREV|NEXT]
... [HOLD] [EXCLUSIVE]
```

- **Format 8, based on position within the set when the record is unknown:**

```
DB-OBTAIN REC|REF IDMSREC SET setname|AREA areaname
... [WHERE SEQUENCE = number|FIRST|LAST|PREV|NEXT]
... [HOLD] [EXCLUSIVE]
```

- **Format 9, based on position within the area:**

```
DB-OBTAIN REC|REF recordname [AREA areaname]
... [WHERE SEQUENCE = number|FIRST|LAST|PREV|NEXT]
... [HOLD] [EXCLUSIVE]
```

DB-OPEN

```
DB-OPEN [ALL] [MODE usagemode] [AREA areaname]
```

DB-PROCESS

```
DB-PROCESS [REF recordname] REC recordname
... [DB-PROCESS-ID name]
... [WHERE keyname operator value]
... [SET setname|AREA areaname] [RESET]
... [HOLD] [EXCLUSIVE]
    Controlled logic block
```

DB-STORE

```
DB-STORE REC recordname
```

Error Handling Flags

Use the following 88-level flags for error handling after executing an IDMS call:

Flag	Status Code	Explanation
AB-ON-REC	0001 through 0306 0308 through 0325 0327 through 0625 1206 through 9999	Any error other than those listed.

Flag	Status Code	Explanation
DUP-ON-REC	0705 0805,1205	Duplicate key.
END-ON-REC	0307	End of set, area, or index reached.
NTF-ON-REC	0326 0626	Record not found.
OK-ON-REC	0000	Successful operation.
POS-ON-REC	All values of AB-ON-REC whose last 2 bytes are: 06 13	Positioning error.
VIO-ON-REC	All values of AB-ON-REC whose last 2 bytes are: 01 02 08 09 10 14 15 23 31	An update would violate rules.

IDM-COMMIT

IDM-COMMIT [ALL]

IDM-CONNECT

IDM-CONNECT REC [recordname] TO setname

IDM-DISCONNECT

IDM-DISONNECT REC [recordname] FROM setname

IDM-IF

IDM-IF SET *setname* [EMPTY|NOT EMPTY]
... [MEMBER|NOT MEMBER] *paragraphname*

IDM-PROTOCOL

-KYWD- 12--*--20---*-----30----*----40---*-----
SYEN IDM-PROTOCOL programmode location

IDM-RETURN

```
IDM-RETURN dataname FROM indexsetname
... [CURRENCY [FIRST|LAST|NEXT|PRIOR]]
... [USING keyfiel] [KEY INTO keyfield]
```

IDM-ROLLBACK

```
IDM-ROLLBACK [CONTINUE]
```

IMS DB

DB-ERASE

- Format 1:

```
DB-ERASE REC segment [FROM dataarea]
... [VIEW pcbname|PCB pcbname]
```

- Format 2, records obtained by path calls:

```
DB-ERASE REC|REF segment1 [FROM dataarea]
... VIEW pcbname|PCB pcbname]
... REC|REF segment2 [FROM dataarea]
... VIEW pcbname|PCB pcbname]
.
.
... REC segmentN [FROM dataarea]
... VIEW pcbname|PCB pcbname]
```

DB-MODIFY

```
DB-MODIFY REC|REF recordname1 [FROM dataarea]
... [VIEW pcbname|PCB pcbname]
.
.
... REC|REF recordnameN [FROM dataarea]
... [VIEW pcbname|PCB pcbname]
```

DB-OBTAIN

- Format 1, unqualified:

```
DB-OBTAIN REC recordname [HOLD] [RESET]
```

- Format 2, qualified:

```
DB-OBTAIN REC recordname
... WHERE fieldname1 operator value
... [AND|OR fieldname2 operator value [FIRST|LAST]
... [INTO dataarea]
.
.
... [AND|OR fieldnameN operator value] [FIRST|LAST]
... [INTO dataarea] [HOLD]]] [RESET]
```

- Format 3, qualified on secondary index values:

```
DB-OBTAIN REC recordname
... WHERE fieldname operator (value1 [ ... valueN])
... [INTO dataarea]
... [FIRST|LAST] [HOLD] [RESET]
```

- Format 4, qualified compound retrieval:

```
DB-OBTAIN REF segmentname1 WHERE fieldname1 operator value
... REC|REF segmentname2 WHERE fieldname2 operator value
.
.
... [REC segmentnameN WHERE fieldnameN operator value
... [FIRST|LAST] [HOLD]] [RESET]
```

- Format 5, retrieve next segment:

```
DB-OBTAIN NEXT[REC] INTO dataarea
...VIEW pcbname|PCB pcbname [HOLD] [RESET]
```

- Format 6, retrieve segment specified in program at run time:

```
MOVE 'segmentname' TO segmentname
DB-OBTAIN IMSREC segmentname FROM dataarea
... VIEW pcbname|PCB pcbname
```

- Format 7, retrieve dependent of current record:

```
DB-OBTAIN REF recordname1 CURRENT
... REC recordname2 [WHERE fieldname operator value]
```

- **Format 8, retrieve segment from PSB with multiple PCBs:**

```
DB-OBTAIN REC recordname
... [WHERE fieldname operator value] VIEW pcbname|PCB pcbname
```

- **Format 9, retrieve qualified or subscripted field:**

```
DB-OBTAIN REC recordname
... WHERE keyname operator fieldname OF dataarea|SUB (number)
```

- **Format 10, retrieve dependent record via concatenated key:**

```
DB-OBTAIN REC recordname CKEYED dataname
```

DB-PROCESS

- **Format 1, key-qualified:**

```
DB-PROCESS REC recordname
... [WHERE keyname operator value [SUB value]
... [OF dataarea]]
... [DB-PROCESS-ID name] [INTO dataarea]
... [HOLD] [RESET] [VIEW pcbname|PCB pcbname]
    Controlled logic block
```

- **Format 2, unqualified:**

```
DB-PROCESS REC recordname
... [DB-PROCESS-ID name] [INTO dataarea]
... [HOLD][RESET] [VIEW pcbname|PCB pcbname]
    Controlled logic block
```

DB-STORE

```
DB-STORE [REC|REF recordname1] [FROM dataarea]
... [VIEW pcbname|PCB pcbname]
... [WHERE fieldname operator value] [REC|REF recordname2]
... [FROM dataarea] [VIEW pcbname|PCB pcbname]
... [WHERE fieldname operator value]
... [SUB number] [OF dataarea]
.
.
... REC recordnameN [FROM dataarea]
... [VIEW pcbname|PCB pcbname]
... [WHERE fieldname operator value] [SUB number] [OF dataarea]
```

Error Handling Fields

IMS provides the following error fields that show how far your call was processed prior to failure:

IM-DB-PCB-SEGLEV	Lowest level in the database for which the requested segment was found.
IM-DB-PCB-SEGNAME	8-character IMS name for the lowest-level segment located.
IM-DB-PCB-KEY-FEED-BACK	Concatenated key information for the path from the root-level to the lowest-level segment found.
IM-DB-PCB-KEY-KFBLEN	Length of data in the IM-DB-PCB-KEY-FEED-BACK field.

Error Handling Flags

Use the following 88-level flags for error handling after executing an IMS DB call:

Flag	Status Code	Explanation
AB-ON-REC	Any not listed below	For any error code not listed below.
DUP-ON-REC	I NI LB	The DB-STORE failed; the new segment would have created a duplicate for a key or sequence field defined as unique.
END-ON-REC	GB	End of database reached.
NTF-ON-REC	GE GB	Requested record not found.
OK-ON-REC	2 spaces, GA GD GK	Everything is OK.
POS-ON-REC	DJ LC LD LE	Positioning error; requested positioning not established.
RTY-ON-REC	GG	Requested record not available. Retry.
VIO-ON-REC	AM DA DX RX IX	An update would violate rules.

Generation Field

IMS provides a field to control program generation:	
&IM-SUPPRESS-DB-CALL	Prevent generation of DB calls; set field to YES in APS CNTL file APSTDBDC.

GSAM Calls

```
IM-CLSE view
IM-OPEN view
IM-OPEN-INP view
IM-OPEN-OUT view
IM-OPEN-OUTA view
IM-OPEN-OUTM view
IM-GN pcbname [ssal [...ssa15]]
IM-GU pcbname ssal [...ssa15]
IM-ISRT pcbname [ssal [...ssa15]]
```

\$IM-FLD

```
$IM-FLD MSDBview fsaname [rootssa]
```

\$IM-FSA

```
$IM-FSA fsaname segment
... field1[/picture/] operator operand1
... [field2[/picture/] operator operand2]
.
.
... [field10[/picture/[ operator operand10]]
```

\$IM-POS

```
$IM-POS DEDBview [SSA]
```

System Service Calls

```

IM-CHKP pcbname checkpointID
... [length1 dataarea1 [... length7 dataarea7]]

IM-XRST pcbname
... [length1 area1 [... length7 area7]]
... [checkpointID maxiolenlength]
IM-CHKP-OSVS pcbname checkpointID
IM-DEQ pcbname deqcharacter
IM-GSCD pcbname
IM-LOG pcbname logcode loglength message
IM-ROLB pcbname [msgarea]
IM-ROLL
IM-STAT-DBAS-FULL pcbname
IM-STAT-DBAS-UNFORMATED pcbname
IM-STAT-DBAS-SUMMARY pcbname
IM-STAT-VBAS-FULL pcbname
IM-STAT-VBAS-UNFORMATED pcbname
IM-STAT-VBAS-SUMMARY pcbname

```

SQL

Built-In Functions

- For DB-DECLARE:

```

DB-DECLARE cursorname [correlname1.]copylibname-REC
... function1[()(expression)[,resultfield[,Y]]]
... function2[()(expression)[,resultfield[,Y]]]
.
.
... functionN[()(expression)[,resultfield[,Y]]]
... WHERE ...

```

- For DB-OBTAIN:

```

DB-OBTAIN REC [correlname1.]copylibname-REC
... function1[()(expression)[,resultfield[,Y]]]
... function2[()(expression)[,resultfield[,Y]]]
.

```

```

... functionN[ ](expression)[,resultfield[,Y][ ]]]
... WHERE ...

```

- For DB-PROCESS:

```

DB-PROCESS REC [correlname1.]copylibname-REC
... [DB-PROCESS-ID name]
... function1[ ](expression)[,resultfield[,Y][ ]]]
... function2[ ](expression)[,resultfield[,Y][ ]]]
...
...
... functionN[ ](expression)[,resultfield[,Y][ ]]]
... WHERE ...

```

DB-CLOSE

```
DB-CLOSE CUR[SOR] cursorname
```

DB-COMMIT

```
DB-COMMIT [HOLD]
```

DB-DECLARE

- Format 1, unqualified, select all columns:

```

DB-DECLARE cursorname copylibname-REC
... [FETCH ONLY] [WITH HOLD] [OPTIMIZE number]
... [UPDATE|ORDER]
... column1 [ASC|DESC] [...columnN [ASC|DESC]]]

```

- Format 2, qualified, select all columns:

```

DB-DECLARE cursorname copylibname-REC
... [FETCH ONLY] [WITH HOLD] [OPTIMIZE number]
... WHERE column operator [[:]altvalue]|column
... [AND|OR column operator [[:]altvalue]|column]
... [UPDATE|ORDER] column1 [ASC|DESC]
... [...columnN [ASC|DESC]]]

```

- **Format 3, select specific columns:**

```
DB-DECLARE cursorname copylibname-REC [DISTINCT]
... [FETCH ONLY] [WITH HOLD] [OPTIMIZE number]
... column1 [(altvalue)] [... columnN [(altvalue)]]
... [WHERE column operator [[:]altvalue]|column
... [AND|OR correlname.]column operator [[:]altvalue]|column]
.
.
... [AND|OR correlname.]column operator [[:]altvalue]|column]]
... [UPDATE|ORDER coll [ASC|DESC] [...columnN [ASC|DESC]]]
```

- **Format 4, join columns from two or more tables:**

```
DB-DECLARE cursorname correlname.copylibname-REC [DISTINCT]
... [FETCH ONLY] [WITH HOLD] [OPTIMIZE number]
... [column1 [(altvalue)] [... columnN [(altvalue)]]]
... [correlname.copylibname-REC
... [column1 [(altvalue)] [... columnN [(altvalue)]]]
.
.
... [WHERE correlname.column operator [[:]altvalue|correlname.column
... [AND|OR correlname.column operator [[:]altvalue|correlname.column]
.
.
... [AND|OR correlname.column operator
[:]altvalue|correlname.column]]
... [ORDER column1 [ASC|DESC] [...columnN [ASC|DESC]]]
```

- **Format 5, specify a Union**

```
DB-DECLARE cursorname copylibname-REC [DISTINCT]
... [FETCH ONLY] [WITH HOLD] [OPTIMIZE number]
... [column1 [(altvalue)] [... columnN [(altvalue)]]]
... [WHERE column operator [[:]altvalue]|column
... [AND|OR column operator [[:]altvalue]|column]
.
.
... [AND|OR column operator [[:]altvalue]|column]]
... UNION [ALL]
DB-OBTAIN REC copylibname-REC
.
.
... [ORDER column1 [ASC|DESC] [...columnN [ASC|DESC]]]
```

DB-ERASE

- Format 1:

```
DB-ERASE REC copylibname-REC
... [WHERE column operator [:]altvalue
... [AND|OR column operator [:]altvalue
.
.
... AND|OR column operator [:]altvalue]]
```

- Format 2:

```
DB-ERASE REC copylibname-REC [WHERE CURRENT [OF]
cursorname]
```

DB-FETCH

```
DB-FETCH CUR[SOR] cursorname [INTO dataname]
```

DB-MODIFY

```
DB-MODIFY REC copylibname-REC
... [column1 [(altvalue)] [... columnN [(altvalue)]]]
... [FROM dataname] [WHERE column1 operator [:]altvalue
.
.
... AND|OR columnN operator [:]altvalue]
... [END[WHERE]]] [WHERE CURRENT [OF] cursorname]
```

DB-OBTAIN

- Format 1, qualified, select all columns:

```
DB-OBTAIN REC copylibname-REC
... WHERE column operator [:]altvalue|column
... [AND|OR column operator [:]altvalue|column]
.
.
... [AND|OR column operator [:]altvalue|column]
... [INTO dataname]
```

- **Format 2, unqualified, select all columns:**

```
DB-OBTAIN REC copylibname-REC [INTO dataname]
```

- **Format 3, join all columns from two tables:**

```
DB-OBTAIN REC correlname1.copylibname-REC
... REC correlnameN.copylibname-REC
... [WHERE correlname.column operator [:]altvalue|correlname.column
... [AND|OR correlname.column operator [:]altvalue|correlname.column]
.
.
... [AND|OR correlname.column operator [:]altvalue|correlname.column]]
```

- **Format 4, select specific columns:**

```
DB-OBTAIN REC copylibname-REC [DISTINCT]
... column1 [(altvalue)] [... columnN [(altvalue)]]
... [WHERE [correlname.]column operator [:]altvalue|column]
```

DB-OPEN

```
DB-OPEN CUR[SOR] cursorname
```

DB-PROCESS

- **Format 1, unqualified, select all columns:**

```
DB-PROCESS REC copylibname-REC
... [DB-PROCESS-ID name] [DB-LOOP-MAX=number]
... [FETCH ONLY] [WITH HOLD] [OPTIMIZE number]
... [UPDATE|ORDER col1 [ASC|DESC] [...columnN [ASC|DESC]]]
... [INTO dataname]
    Controlled logic block
```

- **Format 2, qualified, select all columns:**

```
DB-PROCESS REC copylibname-REC [DB-PROCESS-ID name]
... [FETCH ONLY] [WITH HOLD] [OPTIMIZE number]
... WHERE column operator [[:]]altvalue|column
... [AND|OR column operator [[:]altvalue|column]
.
.
... [AND|OR column operator [[:]altvalue|column]
... [DB-LOOP-MAX=number]
... [UPDATE|ORDER col1 [ASC|DESC] [...columnN [ASC|DESC]]]
```

```
... [INTO dataname]
    Controlled logic block
```

- **Format 3, select specific columns:**

```
DB-PROCESS REC copylibname-REC
... [FETCH ONLY] [WITH HOLD] [OPTIMIZE number]
... [DB-PROCESS-ID name] [DISTINCT]
... column1 [(altvalue)] [... columnN [(altvalue)]]
... WHERE column operator [[:]altvalue]|column
... [AND|OR column operator [[:]altvalue]|column]
.
.
... [AND|OR col operator [[:]altvalue]|col]
... [DB-LOOP-MAX=number]
... [UPDATE|ORDER column1 [ASC|DESC] [...columnN [ASC|DESC]]]
    Controlled logic block
```

- **Format 4, join columns from two or more tables:**

```
DB-PROCESS REC correlname.copylibname-REC
... [DB-PROCESS-ID name] [DISTINCT]
... [FETCH ONLY] [WITH HOLD] [OPTIMIZE number]
... [column1 [(altvalue)] [... columnN [(altvalue)]]]
... REC correlname.copylibname-REC
... [column1 [(altvalue)] [... columnN [(altvalue)]]]
.
.
... [WHERE correlname.column operator [[:]altvalue]|correlname.column]
... [AND|OR correlname.column operator [[:]altvalue]|correlname.column]
.
.
... [AND|OR correlname.column operator [[:]altvalue]|correlname.column]
... [DB-LOOP-MAX=number]
... [ORDER column1 [ASC|DESC] [...columnN [ASC|DESC]]]
    Controlled logic block
```

- **Format 5, specifying a UNION:**

```
DB-PROCESS REC copylibname-REC
... [DB-PROCESS-ID name] [DISTINCT]
... [FETCH ONLY] [WITH HOLD] [OPTIMIZE number]
... [column1 [(altvalue)] [... columnN [(altvalue)]]]
... [WHERE column operator [[:]altvalue]|column]
... [AND|OR column operator [[:]altvalue]|column]
.
.
... [AND|OR column operator [[:]altvalue]|column]]
... [DB-LOOP-MAX=number] UNION [ALL]
```

```
DB-OBTAIN REC copylibname-REC
.
.
... [ORDER column1 [ASC|DESC] [...columnN [ASC|DESC]]]
      Controlled logic block
```

DB-ROLLBACK

```
DB-ROLLBACK
```

DB-STORE

- **Format 1:**

```
DB-STORE REC copylibname-REC
... [column1 [(altvalue)] [... columnN [(altvalue)]]]
... [FROM dataname]
```

- **Format 2:**

```
DB-STORE REC copylibname-REC
... [column1 [(altvalue)] [... columnN [(altvalue)]]]
... [DB-OBTAIN REC copylibname-REC column1 [... columnN]
... WHERE column1 operator [:]altvalue
... [AND|OR column2 operator [:]altvalue]
.
.
... [AND|OR columnN operator [:]altvalue] ]
```

Error Handling Flags

Use the following 88-level flags for error handling after executing SQL calls:

Flag	Meaning
AB-ON-REC	Any error not listed in this table.
DB2-DEADLOCK	DB-PROCESS calls check this status to ensure the cursor is not already closed before closing it. SQL closes the cursor if the database is locked.
DUP-ON-REC	DB-STORE failed because the row already exists; duplicates are not allowed.

Flag	Meaning
END-ON-REC	End of table or cursor set reached.
NTF-ON-REC	Requested row not found.
OK-ON-REC	Operation successful.
RI-ON-REC	Referential Integrity check successful (corresponds to SQLCODE -532 to -530).

Error Handling Field

Use the following generated field to indicate whether the associated host variable has been assigned a null value:

```
01  IND-cursorname|IND-recorename
05  IND-column
```

Joins

- With DB-DECLARE:

```
DB-DECLARE cursorname correlname1.copylibname-REC [DISTINCT]
... [column1 [... columnN]]|[NONE]
.
... correlnameN.copylibname-REC [column1 [... columnN]]|[NONE]
.
... [WHERE correlname.column1 operator [:]value|correlname.column2
... [AND|OR correlname.column3 operator [:]value|correlname.column4
.
... AND|OR correlname.columnN operator [:]value|correlnameN]]
... [ORDER column1 [ASC|DESC] [...columnN [ASC]]]
```

- With DB-OBTAIN:

```
DB-OBTAIN REC correlname1.copylibname-REC [DISTINCT]
... [column1 [... columnN]]|[NONE]
.
.
.
... REC correlnameN.copylibname-REC [column1 [... columnN]]|[NONE]
... [WHERE correlname.column1 operator [:]value|correlname.column2
... [AND|OR correlname.column3 operator [:]value|correlname.column4
.
.
.
```



```

.
... AND|OR correlname.columnN operator
[:]value|correlname.columnN]]

```

- **With DB-PROCESS:**

```

DB-PROCESS REC correlname1.copylibname-REC
... [DB-PROCESS-ID name] [DISTINCT]
... [column1 [... columnN]]|[NONE]
.
.
.
... REC correlnameN.copylibname-REC
... [column1 [... columnN]]|[NONE]

... [WHERE correlname.column1 operator [:]value|correlname.column2
... [AND|OR correlname.column3 operator [:]value|correlname.column4
.
.
.
... AND|OR correlname.columnN operator [:]value|correlname.columnN]]

... [DB-LOOP-MAX=number]
... [ORDER column1 [ASC|DESC] [...columnN [ASC|DESC]]]
    Controlled logic block

```

Special Registers

Use the following special registers in DB-DECLARE, DB-OBTAIN, and DB-PROCESS calls:

```

CURRENT DATE
CURRENT TIME
CURRENT TIMESTAMP
CURRENT TIMEZONE

```

UNION

- **With DB-DECLARE:**

```

DB-DECLARE cursorname [correlname1.]copylibname-REC
.
.
.
... UNION [ALL]

```

```

DB-OBTAIN REC copylibname-REC
.
.
.
... [ORDER column1 [ASC|DESC] [...columnN [ASC|DESC]]]

```

- With DB-PROCESS:

```

DB-PROCESS REC [correlname1.]copylibname-REC
.
.
.
... UNION [ALL]
DB-OBTAIN REC copylibname-REC
.
.
.
... [ORDER column1 [ASC|DESC] [...columnN [ASC|DESC]]]

```

VSAM Batch

DB-CLOSE

- Format 1:

```
DB-CLOSE FILE filename1 [ ... filenameN]
```

- Format 2:

```
DB-CLOSE FILE ALL
```

DB-ERASE

- Format 1, key-qualified:

```

DB-ERASE REC recordname
... WHERE primarykeyname = value [SUB value]
... [OF dataarea]

```

- Format 2, unqualified:

```
DB-ERASE REC recordname
```

DB-MODIFY

DB-MODIFY REC recordname [FROM dataarea]

DB-OBTAIN

- Format 1, sequential:

DB-OBTAIN REC recordname [VIEW keyname] [INTO dataarea] [RESET]

- Format 2, direct:

DB-OBTAIN REC recordname WHERE keyname operator value [SUB value]
... [OF dataarea] [INTO dataarea]

- Format 3, positional:

DB-OBTAIN REC recordname WHERE keyname operator value [SUB value]
... [OF dataarea]

DB-OPEN

- Format 1:

DB-OPEN FILE filename ...[filenameN] MODE option

- Format 2:

DB-OPEN FILE ALL MODE option

DB-PROCESS

- Format 1, qualified:

DB-PROCESS REC *recordname* WHERE *keyname operator value*
... [DB-PROCESS-ID name] [SUB value] [OF dataarea]
... [INTO dataarea]
Controlled logic block

- Format 2, unqualified:

DB-PROCESS REC recordname [VIEW keyname]
... [DB-PROCESS-ID name] [INTO dataarea] [RESET]
Controlled logic block

DB-STORE

```
DB-STORE REC recordname [FROM dataarea]
```

Error Handling Flags

Use the following 88-level flags for error handling after executing SQL calls:

Flag	Status Code	Explanation
OK-ON-REC	00	Successful operation.
DUP-ON-REC	02	Duplicate key; duplicates allowed.
END-ON-REC	10	End of file.
INV-ON-REC	20 21 22 23 24	Invalid key condition.
IVD-ON-REC	22	Duplicate key; not allowed.
NTF-ON-REC	23	Record not found.
AB-ON-REC	30 34 90 91 92 93 94 95 96 97	Abnormal condition.

Fields

Use APS-defined data fields and S-COBOL flags in your program.

Name	Associated Calls	Description
APS-END-PROCESS	DB-PROCESS	Flag that terminates a DB-PROCESS loop.
<i>ddname</i> -RRN	DB-OBTAIN DB-PROCESS DB-ERASE DB STORE	Field that controls relative record number (RRN) of a retrieved or stored RRDS file record; generates value for the RELATIVE KEY clause of the SELECT statement.
RESET-OBTAIN	DB-OBTAIN DB-PROCESS	Flag that resets browse to beginning of file. Example: <i>TRUE RESET-OBTAIN</i>

VSAM Online

DB-ERASE

- Format 1, key-qualified:

```
DB-ERASE REC recordname WHERE primarykeyname = value  
... [SUB value] [OF dataarea] [KLEN value]  
... [SYSID systemname] [DDN ddname]
```

- Format 2, unqualified:

```
DB-ERASE REC recordname
```

DB-FREE

```
DB-FREE REC recordname [ALL [VIEW keyname] [ENDBR] [UNLOCK]  
... [REQID number] [SYSID systemname] [DDN ddname]
```

DB-MODIFY

```
DB-MODIFY REC recordname [FROM dataarea]  
... [SYSID systemname] [DDN ddname]
```

DB-OBTAIN

- Format 1, sequential:

```
DB-OBTAIN REC recordname [VIEW keyname] [INTO dataarea]  
... [HOLD] [PREV] [REQID number] [RESET]  
... [SYSID systemname] [DDN ddname]
```

- Format 2, direct:

```
DB-OBTAIN REC recordname WHERE keyname operator value  
... [SUB value] [OF dataarea] [INTO dataarea] [KLEN value]  
... [HOLD]  
... [REQID number] [SYSID systemname] [DDN ddname]
```

- Format 3, positional:

```
DB-OBTAIN REF recordname WHERE keyname operator value
... [SUB value] [OF dataarea] [KLEN value] [RESETBR]
... [REQID number] [SYSID systemname] [DDN ddname]
```

DB-PROCESS

- Format 1, key-qualified:

```
DB-PROCESS REC recordname WHERE keyname operator value
... [DB-PROCESS-ID name] [SUB value] [OF dataarea]
... [INTO dataarea] [KLEN value] [HOLD] [PREV]
... [REQID number] [SYSID systemname] [DDN ddname]
    Controlled logic block
```

- Format 2, unqualified:

```
DB-PROCESS REC recordname [DB-PROCESS-ID name]
... [INTO dataarea]
... [HOLD] [PREV] [RESET] [REQID number]
... [SYSID systemname]
... [VIEW keyname] [DDN ddname]
    Controlled logic block
```

DB-STORE

```
DB-STORE REC recname [FROM dataarea] [SYSID systemname]
... [DDN ddname]
```

Error Handling Flags

Use the following 88-level flags for error handling after executing VSAM online calls:

VSAM Flag	EIBRCODE Flag	ISI-Errors/ Exceptional Condition	Explanation
AB-ON-REC	DSIDERR ILLOGIC IOERR LENGERR	DSIDERR ILLOGIC IOERR LENGERR	Abnormal condition.

VSAM Flag	EIBRCODE Flag	ISI-Errors/ Exceptional Condition	Explanation
	NOSPACE NOTOPEN SYSIDERR	NOSPACE NOTOPEN SYSIDERR	
DUP-ON-REC	DUPKEY	DUPKEY	Duplicate key; allowed.
END-ON-REC	ENDFILE	ENDFILE	End of file.
INV-ON-REC	NOTFND DUPREC	NOTFND DUPREC	Invalid key condition.
IRQ-ON-REC	INVREQ	INVREQ	Invalid request.
IVD-ON-REC	DUPREC	DUPREC	Duplicate key; not allowed.
NTF-ON-REC	NOTFND	NOTFND	Record not found.
OK-ON-REC	N/A	N/A	Successful operation.

Fields

Use APS-defined data fields and S-COBOL flags in your program.

Name	Associated Calls	Description
<i>APS-shortrecname-VAR</i>	DB-OBTAIN DB-STORE DB-MODIFY DB-PROCESS DB-ERASE	Field that contains actual record length after a successful record retrieval (direct or sequential for DB-OBTAIN and DB-PROCESS). Supplies value to <i>dataarea</i> in CICS LENGTH option. <i>Shortrecname</i> comes from the subschema definition; it is the REC card SHORT keyword.
<i>APS-END-PROCESS</i>	DB-PROCESS	Flag that terminates DB-PROCESS loop.

Name	Associated Calls	Description
<i>APS-VSAM-NUMREC</i>	DB-ERASE	Field that contains number of records deleted after key-qualified ERASE with partial key length specified.
<i>ddname-APS-KEYnumber</i>	DB-OBTAIN DB-PROCESS	Field that contains APS-generated key name for use in skip-sequential processing. <i>Ddname</i> is the subschema file.
<i>ddname-RBA</i>	DB-OBTAIN DB-PROCESS DB-STORE	Field that contains relative byte address (RBA) of a retrieved, stored ESDS file record; supplies value to CICS RIDFLD option.
<i>ddname-RRN</i>	DB-OBTAIN DB-PROCESS DB-STORE	Field that contains relative record number (RRN) of a retrieved or stored RRDS file record; supplies value to CICS RIDFLD option.
<i>RESET-OBTAIN</i>	DB-OBTAIN DB-PROCESS	Flag that resets browse to beginning of file. When used with PREV, resets to end of file.

6 Data Communication Calls

This chapter contains information on coding Data Communication Calls, including CICS, IMS DC, ISPF Dialog, and ISPF Prototyping.

Coding Calls

Observe the following when coding database calls:

- Code the command using COBOL indentation.
- Arguments can be one of the following:
 - Variable
 - 1- to 7-digit number
 - Literal string delimited by single or double quotes
- Separate each command component with a space, unless indicated otherwise in the syntax.
- Some arguments are positional. Code positional arguments in the order shown in the syntax for each command.
- To omit a positional argument, code an asterisk (*) in its place (unless it is the last argument).
- A command can be continued on as many as 101 subsequent lines, by coding an ellipsis followed by a space. Break a command for continuation at any blank space, but do not break a parameter.
- The plus symbol (+) serves as a text separator.
- Comments:
 - Code anywhere in your program:

```
-KYWD- 12-*----20---*--
/*      commentline
```

- Code in Procedure Division only:

```
-KYWD- 12-*----20---*----
/*commentline
```

- Continuation is an ellipsis followed by a space (...) and must continue on the immediate subsequent line.

CICS

ATTR

```
[TP-]ATTR screenname attribute1[+attribute2...]
... fieldname[(subscript)][+fieldname[(subscript)] ...]
```

Valid attributes are: *BRT, NORM, DARK, MDTON, MDTOF, NUM[LOCK], NUMOFF, POS, [UN]PROT, ASKIP, DET, DETOFF, DEFCOL, RD, BL, PK, YL, GN, TQ, NU, [NO]BLINK, [NO]RVID, [NO]UNDER.*

CIC-ADDRESS

```
CIC-ADDRESS option(linkdataname) [option(linkdataname) ...]
```

CIC-ASSIGN

```
CIC-ASSIGN CICSoption(dataarea) [CICSoption(dataarea)....]
... [ERROR(errorpara)]
```

CIC-CANCEL

```
CIC-CANCEL [REQID(name)] [TRANSID(name)]
... [SYSID(name)] [ERROR(errorpara)]
```

CIC-DELAY

```
CIC-DELAY [ REQID(name)] [ INTERVAL(hhmmss)|TIME(hhmmss)]
... [ERROR(errorpara)]
```

CIC-DELETEQ

```
CIC-DELETEQ-TD|TS QUEUE(name) [SYSID(name)] [ERROR(errorpara)]
```

CIC-FREEMAIN

```
CIC-FREEMAIN DATA(linkdataname)
```

CIC-GETMAIN

```
CIC-GETMAIN SET(linkdataname) LENGTH(value)|FLENGTH(value)
... [INITIMG(value)] [ERROR(errorpara)]
```

CIC-LOAD

```
CIC-LOAD PROGRAM(name) [SET(linkdataname)]
... [LENGTH(dataarea)]|[FLENGTH(dataarea)]
... [ENTRY(pointref)] [HOLD] [ERROR(errorpara)]
```

CIC-READQ-TD

```
CIC-READQ-TD QUEUE(name) INTO(dataarea)|SET(linkdataname)
... [LENGTH(dataarea)] [SYSID(name)] [ERROR(errorpara)]
```

CIC-READQ-TS

```
CIC-READQ-TS QUEUE(name) INTO(dataarea)|SET(linkdataname)
... LENGTH(dataarea) NUMITEMS(dataarea) [ITEM(value)|NEXT]
... [SYSID(name)] [ERROR(errorpara)]
```

CIC-RELEASE

CIC-RELEASE PROGRAM(name) [ERROR(errorpara)]

CIC-SCHEDULE-PSB

CIC-SCHEDULE-PSB

CIC-SEND-TEXT

CIC-SEND-TEXT FROM(dataarea) LENGTH(value)
... [CISOptions] [ERROR(errorpara)]

CIC-SERVICE-RELOAD

CIC-SERVICE-RELOAD linkdataname

CIC-START

CIC-START TRANSID(name) [INTERVAL(hhmmss) | TIME(hhmmss)]
... [CISOptions] [ERROR(errorpara)]

CIC-TERM-PSB

CIC-TERM-PSB

CIC-WRITEQ-TD

CIC-WRITEQ-TD QUEUE(name) FROM(dataarea)
... [LENGTH(value)] [SYSID(name)] [ERROR(errorpara)]

CIC-WRITEQ-TS

```
CIC-WRITEQ-TS QUEUE(name) FROM(dataarea) LENGTH(value)
... [SYSID(name)] [ITEM(dataarea) [REWRITE] [CICSOptions]
... [NOSUSPEND] [ERROR(errorpara)]
```

CLEAR

```
[TP-|SC-]CLEAR screenname
```

CLEAR-ATTRS

```
[TP-]CLEAR-ATTRS screenname
```

Error Handling Flags

Use the following 88-level flags for error handling after executing any CICS call:

CBIDERR	FUNCERR	NOPASSBKRD	SEGIDERR
DISABLED	IGREQCD	NOPASSBKWR	SELNERR
DSIDERR	ILLOGIC	NOSPACE	SESSBUSY
DSSTAT	INVERRTERM	NOSTG	SESSIONERR
DUPKEY	INVMPsz	NOTALLOC	SIGNAL
DUPREC	INVREQ	NOTFND	SYSBUSY
ENDDATA	INVTsREQ	NOTOPEN	SYSIDERR
ENDFILE	IOERR	PGMIDERR	TERMidERR
ENDINPT	ISCINVREQ	QBUSY	TRANSIDERR
ENQBUSY	ITEMERR	QIDERR	UNEXPIN
ENVDEFERR	JIDERR	QZERO	WRBRK
EODS	LENGERR	RDATT	
EOF	MAPFAIL	RETPAGE	
EXPIRED	NOJBUFSz	RTEFAIL	

Invocation Mode Flags

Use the following 88-level fields provided by APS to indicate program invocation mode.

```
TP-INVOCAATION-MODE          PIC X(01).
88  TP-TRANSID-INVOKEO      VALUE 'T'.
88  TP-PROGRAM-INVOKEO      VALUE 'P'.
88  TP-SCREEN-INVOKEO       VALUE 'S'.
88  TP-LINK-INVOKEO         VALUE 'L'.
```

LINK

- Format 1, linking to an APS program:

```
[TP-]LINK programname [errorpara]
... [DLIUIB pcbname [pcbname] ...]
... [userparm [userparm]...]
... [COMMAREA(dataarea) LENGTH(value)]|[NOCA]
```

- Format 2, linking to a non-APS program:

```
[TP-]LINK programname(NONAPS)[errorpara]
... [COMMAREA(dataarea) LENGTH(value)]|[NOCA]
```

NTRY

```
NTRY|ENTR screenname[(mapsetname)] [errorpara] [RETRY|NORETRY]
```

PF Key Values

Use the following PF key 88-levels generated by APS.

88-level PF-KEY	Value	88-levelPF-KEY	Value
ENTER-KEY	'''	PF8	'8'
CLEAR-KEY	'_'	PF9	'9'
PEN	'='	PF10	':'
OPID	'W'	PF11	'#'
MSRE	'X'	PF12	'@'

88-level PF-KEY	Value	88-levelPF-KEY	Value
STRF	'H'	PF13	'A'
TRIG	'"'	PF14	'B'
PA1	'%'	PF15	'C'
PA2	'>'	PF16	'D'
PA3	'.'	PF17	'E'
PF0 PF00	'"'	PF18	'F'
PF1	'1'	PF19	'G'
PF2	'2'	PF20	'H'
PF3	'3'	PF21	'I'
PF4	'4'	PF22	'['
PF5	'5'	PF23	'.'
PF6	'6'	PF24	'<'
PF7	'7'		

RESET-PFKEY

[TP-]RESET-PFKEY keyvalue

SCRNLIST

[TP-]SCRNLIST screenname1 [... screenname12]
... [MAPSET(mapsetname)] [LINKAGE] [REDEFINE|NOREDEF]

SEND

[TP-]SEND screen[(mapsetname)] [errorpara]
... [TRANSID(name)] [NORETURN] [NOERASE]
... [CICSoption [CICSoption] ...]

TERM

[TP-]TERM

TP-BACKOUT

```
TP-BACKOUT [ABORT[(name)]|NOABORT]
```

TP-LINKAGE

```
TP-LINKAGE linkdataname[/copybookname/macrofilename]
... [linkdataname[/copybookname/macrofilename] ...]
```

TP-NULL

```
TP-|SC-NULL screenname
```

TP-PERFORM

- Format 1, performing a paragraph:

```
TP-PERFORM paragraphname
```

- Format 2, performing a paragraph that passes arguments:

```
TP-PERFORM paragraphname actualarg1 [... actualarg8]
.
.
paraname ([+|-]formalarg1 [... [+|-]formalarg8)
```

XCTL

- Format 1, transferring to an APS program:

```
[TP-]XCTL programname [errorpara] [LENGTH(value)]
... [DLIUIB pcbname [pcbname ...] [userparm [userparm]
...]
```

- Format 2, transferring to a non-APS program:

```
[TP-]XCTL programname(NONAPS) [errorpara] [LENGTH(value)]
```

IMS DC

ATTR

```
[TP-]ATTR screenname attribute1[+attribute2...]  
... fieldname[(subscript)][+fieldname[(subscript)] ...]
```

Valid attributes are: *BRT, NORM, DARK, NUM[OFF], POS, [UN]PROT, ASKIP, DET, DETOFF, DEFCOL, RD, BL, PK, YL, GN, TQ, NU, [NO]BLINK, [NO]RVID, [NO]UNDER.*

CLEAR

```
[TP-|SC-]CLEAR screenname
```

CLEAR-ATTRS

```
[TP-]CLEAR-ATTRS screenname
```

Error Handling Flags

Use the following 88-level flags for error handling after executing any IMS DC call:

Flag	Status Code	Explanation
AB-ON-DC-CALL	CH X1 X8	Category 5 status code returned; call is not completed.
FP-ERR	FF FH FS FV	Category 3 status code returned. Fast Path error occurred; call is completed.
NO-MORE-MSGs	QC	Category 3 status code returned on the TP call; no more input messages exist.

Flag	Status Code	Explanation
NO-MORE-SEGS	QD	Category 3 status code returned; no more segments exist for this message.
SEG-NOT-FOUND	GE	Category 1 status code returned; segment not found.
OK-ON-DC-CALL	2 spaces, CC CE CF CG CI CJ CK CL FD FW FF FH FS FV GE QC QD	Categories 1 and 2 status codes returned; processing proceeds.
SEC-VIO	A4 FI	Category 4 status code returned; security violation occurred; call is not completed.
SPA-IO-ERR	XA XB XE XF XG X1 X2 X3 X4 X5 X6 X7 X8 X9	Categories 4 and 5 status codes returned; SPA error occurred; call is not completed.
TP-PGM-ERR	AA AB AD AL AP AT AY AZ A1 A2 A3 A4 A5 A6 A7 A8 A9 CA CB CD QE QH	Category 4 status code returned. Programming error occurred; call is not completed.

\$IM- Calls

```
$IM-CHNG altview [destination]
$IM-CMD [PCBname] [msgarea]
$IM-GCMD [PCBname] [msgarea]
$IM-GN PCBname ssal [... ssal5]
$IM-GU PCBname ssal [... ssal5]
$IM-ISRT [PCBname]altview] ssal [... ssal5]
$IM-PURG [PCBname] [msgarea] [mod]
```

Invocation Mode Flags

Use the following 88-level fields provided by APS to indicate program invocation mode.

```
TP-INVOCATION-MODE          PIC X.
88  TP-TRANSID-INVOKED      VALUE 'T'.
88  TP-PROGRAM-INVOKED      VALUE 'P'.
88  TP-SCREEN-INVOKED       VALUE 'S'.
```

LINK

```
[TP-]LINK subprogram [errorpara] [argument1 ... argument36]
```

MSG-SW

```
[TP-]MSG-SW trancode|programname|dataname [errorpara]
... [screenname|recordname] [keyword[+keyword]...]
```

NTRY

- Format 1:

```
NTRY|ENTR
```
- Format 2:

```
NTRY|ENTR screenname [errorpara] [RETRY|NORETRY]
```
- Format 3:

```
NTRY|ENTR recordname [errorpara] [RETRY|NORETRY] *RECORD
```

PF Key Values

Use the following PF key 88-levels generated by APS.

88-level PF key	Value	88-level PF key	Value
ENTER-KEY	' '	PF11	'B'

88-level PF key	Value	88-level PF key	Value
NO-KEY-USED	LOW-VALUES	PF12	'C'
PF0 PF00	' '	PF13	'D'
PF1 PF01	'1'	PF14	'E'
PF2 PF02	'2'	PF15	'F'
PF3 PF03	'3'	PF16	'G'
PF4 PF04	'4'	PF17	'H'
PF5 PF05	'5'	PF18	'I'
PF6 PF06	'6'	PF19	'J'
PF7 PF07	'7'	PF20	'K'
PF8 PF08	'8'	PF21	'L'
PF9 PF09	'9'	PF22	'M'
PF10	'A'	PF23	'N'
		PF24	'O'

RESET-PFKEY

[TP-]RESET-PFKEY keyvalue

SCRNLIST

- Format 1:
[TP-]SCRNLIST *screenname1* [/*screenname2*[.../*screenname40*]]
- Format 2:
[TP-]SCRNLIST *screenname1* [*screenname2*[... *screenname40*]

SEND

[TP-]SEND *screenname*|*recordname* [*errorpara*] [*lterm*]
... [*keyword*[+*keyword*] ...]

System Service Calls

```

IM-CHKP PCBname checkpointID
... [length1 dataareal [... length7 dataarea7]]
IM-XRST PCBname
... [length1 area1 [... length7 area7]]
... [checkpointID maxiolenlength]
IM-CHKP-OSVS PCBname checkpointID
IM-DEQ PCBname deqcharacter
IM-GSCD PCBname
IM-LOG PCBname logcode loglength message
IM-ROLB PCBname [msgarea]
IM-ROLL
IM-STAT-DBAS-FULL PCBname
IM-STAT-DBAS-UNFORMATED PCBname
IM-STAT-DBAS-SUMMARY PCBname
IM-STAT-VBAS-FULL PCBname
IM-STAT-VBAS-UNFORMATED PCBname
IM-STAT-VBAS-SUMMARY PCBname

```

TERM

```
[TP-]TERM
```

TP-BACKOUT

```
TP-BACKOUT [ABORT|NOABORT]
```

TP-LINKAGE

```

TP-LINKAGE linkdataname[/copybookname/macrofilename]
... [linkdataname[/copybookname/macrofilename] ...]

```

TP-NULL

```
TP-[SC-NULL screenname
```

TP-PERFORM

- Format 1, performing a paragraph:
`TP-PERFORM paragraphname`
- Format 2, performing a paragraph that passes arguments:
`TP-PERFORM paragraphname`
`... actualargument1 [... actualargument8]`
`.`
`.`
`paraname ([+|-]formalargument1 [... [+|-]formalargument8)`

ISPF Dialog

ATTR

```
[TP-]ATTR screenname attributel[+attribute2...]  
... fieldname[(subscript)][+fieldname[(subscript)] ...]
```

Valid attributes are: *BRT, NORM, DARK, NUM[OFF], POS, [UN]PROT, ASKIP, DET, DETOFF, DEFCOL, RD, BL, PK, YL, GN, TQ, NU, [NO]BLINK, [NO]RVID, [NO]UNDER.*

CLEAR

```
[TP-|SC-]CLEAR screenname
```

CLEAR-ATTRS

```
[TP-]CLEAR-ATTRS screenname
```

DLG-ISPEXEC

```
DLG-ISPEXEC commandproceduresyntax
```

DLG-ISREDIT

DLG-ISREDIT *commandproceduresyntax*

DLG-SETMSG

- Format 1, SETMSG definition:

DLG-SETMSG [*erroridentifier*|*messageID*]

- Format 2, SETMAG definition for execution by Format 3:

DLG-SETMSG [SHORT '*shortmessagetext*'] [LONG '*longmessagetext*']
 ... [ALARM 'YES'|'NO'] [HELP '*helppanelname*']

- Format 3, SETMSG execution:

DLG-SETMSG *erroridentifier*
 ... [SHORT '*shortmessagetext*'] [LONG '*longmessagetext*']
 ... [ALARM 'YES'|'NO'] [HELP '*helppanelname*']

DLG-VCOPY

DLG-VCOPY [COBOLlevel] COBOLvariable [[FROM] dialogvariable]
 ... PIC COBOLpicture|LEN value [GENONLY]

DLG-VDEFINE

DLG-VDEFINE [COBOLlevel] COBOLvariable [[AS] dialogvariable]
 ...PIC COBOLpicture|LEN value [GENONLY]

DLG-VDELETE

DLG-VDELETE dialogvariable|*

DLG-VREPLACE

```
DLG-VREPLACE [COBOLlevel] COBOLvariable
... [[INTO] dialogvariable]
... PIC COBOLpicture|LEN value [GENONLY]
```

DLG-VRESET

```
DLG-VRESET
```

Invocation Mode Flags

Use the 88-level fields provided by APS to indicate program invocation mode.

```
TP-INVOCATION-MODE          PIC X(01).
88  TP-TRANSID-INVOKED      VALUE 'T'.
88  TP-PROGRAM-INVOKED     VALUE 'P'.
88  TP-SCREEN-INVOKED       VALUE 'S'.
88  TP-LINK-INVOKED         VALUE 'L'.
```

LINK

- Format 1, CALL format:

```
[TP-]LINK programname [errorpara]
... [userparm [userparm]...] [COMMAREA]
```
- Format 2, SELECT format:

```
[TP-]LINK programname [errorpara] [options]
```

NTRY

```
NTRY|ENTR screenname [errorpara] [RETRY|NORETRY]
... [CANCEL|RETURN] [dataareas]
```


PF Key Values

Use the following PF key 88-levels generated by APS.

88-level PF key	Value	88-level PF key	Value
ENTER-KEY	' '	PF13	'PF13'
PF1 PF01	'PF01'	PF14	'PF14'
PF2 PF02	'PF02'	PF15	'PF15'
PF3 PF03	'PF03'	PF16	'PF16'
PF4 PF04	'PF04'	PF17	'PF17'
PF5 PF05	'PF05'	PF18	'PF18'
PF6 PF06	'PF06'	PF19	'PF19'
PF7 PF07	'PF07'	PF20	'PF20'
PF8 PF08	'PF08'	PF21	'PF21'
PF9 PF09	'PF09'	PF22	'PF22'
PF10	'PF10'	PF23	'PF23'
PF11	'PF11'	PF24	'PF24'
PF12	'PF12'		

RESET-PFKEY

[TP-]RESET-PFKEY keyvalue

SCRNLIST

- Format 1:
[TP-]SCRNLIST screenname1[(LK)] [... screenname40[(LK)]]
- Format 2:
[TP-]SCRNLIST screenname1[... screenname40]... [LINKAGE]

SEND

[TP-]SEND screen [errorpara] [CONTINUE|NOCONTINUE]

TERM

```
TERM call:ISPF DialogTP-TERM call:ISPF Dialog[TP-]TERM
```

TP-LINKAGE

```
TP-LINKAGE linkdataname[/copybookname/macrofilename]  
... [linkdataname[/copybookname/macrofilename] ...]
```

TP-NULL

```
TP-|SC-NULL screenname
```

TP-PERFORM

- Format 1, performing a paragraph:

```
TP-PERFORM paragraphname
```

- Format 2, performing a paragraph that passes arguments:

```
TP-PERFORM paragraphname  
... actualargument1 [... actualargument8]  
.  
.  
paraname ([+|-]formalargument1 [... [+|-]formalargument8)
```

ISPF Prototyping

ATTR

```
[TP-]ATTR screenname POS fieldname[(subscript)]
```

LINK

```
[TP-]LINK programname [errorpara] [COMMAREA(dataarea)
... LENGTH(value)]|[NOCA]
```

MSG-SW

```
[TP-]MSG-SW trancode|programname|dataname [errorpara]
... [screenname]
```

NTRY

```
NTRY|ENTR screenname [RETRY|NORETRY] [CANCEL|RETURN]
... [dataareas]
```

RESET-PFKEY

```
[TP-]RESET-PFKEY keyvalue
```

SCRNLIST

Any syntax valid in the other DC environments

SEND

- Format 1:

```
[TP-]SEND screen[(mapsetname)] [errorpara]
... [TRANSID(name)] [NORETURN] [NOERASE]
... [CICOption [CICOption] ...]
```

- Format 2:

```
[TP-]SEND screen[(mapsetname)] [errorpara] [lterm]
```

- Format 3:

```
[TP-]SEND screen [errorpara] [CONTINUE|NOCONTINUE]
```

TERM

[TP-]TERM

TP-PERFORM

- Format 1, performing a paragraph:
`TP-PERFORM paragraphname`
- Format 2, performing a paragraph that passes arguments:
`TP-PERFORM paragraphname`
`... actualargument1 [... actualargument8]`
`.`
`.`
`paraname ([+|-]formalargument1 [... [+|-]formalargument8)`

XCTL

- Format 1:
`[TP-]XCTL programname [LENGTH(value)]`
 - Format 2:
`[TP-]XCTL programname(NONAPS) [LENGTH(value)]`
-

7 Data Element Facility

This chapter the screens of the Data Element Facility, and its library administration.

Screens

Context Table Maintenance

Add or modify context entries; enter up to 10 different contexts for a field. To define a context, enter context name in the Context field and its description in the Short Description field.

D in the selection field deletes a context. If any field with the given context is used in a screen, you cannot delete that context, remove the relevant fields from all application screens, and then delete the context.

Data Element Administration

Allow developers to select only data elements defined in the Data Element Library, and prevent them from creating I/O fields by entering YES in the Data Consistency Option field.

Data Element Facility

Create, import, or modify screen field definitions, and access other screens for defining their usage.

Enter indicated option in the Option field and enter values in other screen fields as specified.

Add/modify field definitions

Blank in the Option field to transfer to Screen Field Definition; enter value for Field Name or blank to select from member list; enter value in the Context field, if applicable.

Assign field edits

Option FE to transfer to Field Edits Menu; enter value for Field Name or blank to select from member list; enter value in the Context field, if applicable.

Copy screen field

Option C; value in the Field Name field; enter the new field name in the Newname field; the context of source field in the Context field (optional); enter the context for the new field in the New Context field (optional).

Data Consistency Option

Option Admin to transfer to Data Element Administration to set data consistency option.

Define contexts

Option Context to transfer to Context Table Maintenance.

Delete a screen field

Option D; value for Field Name or blank to select from member list.

Display all contexts

Option CONLIST.

Generate cross-reference for field/screens

Option W; value in the Field Name field or blank to select from the member list; enter the context to report on in the Context field (optional); enter YES in the Batch Execution Option to produce the report in batch mode, NO for online.

Import screen fields

Option I; screen to be imported in the Screen Name field or blank to select from the member list; enter YES or NO in the Replace Like-Named Fields.

Print data element report

Option P; value for the Field Name or blank for all fields; enter the name in the Context field (optional) to report on a field or fields with a specific context or enter * for all contexts.

Rename screen field

Option R; current field name in the Field Name field; enter the new field name in the Newname field; enter current context in the Context field (optional); enter the context for the new field in the New Context field (optional).

Reorganize index structure

Option APSBUILDI.

Field/Screen Cross Reference

The APS Administrator can view a list of screens that reference a specified field here. After a global field is changed by the APS Administrator, this screen shows all screens that use the definition. Generate those screens by typing G next to the applicable screens, or generate all screens by entering GEN in the Command field.

Screen Field Definition

Create or modify a global screen field and its attributes.

Field	Description and Values
Screen Field Name	1-16 characters.
Context	Context in which the field will be used on the screen.
Field Length	Numeric value from 1 through 131.
Text Prompt	Text prompt associated with the field (1-40 characters).
Initial Value	Initial value for the field.
Field Attributes	For attribute information, see <i>See Field Attributes</i> .

Data Element Library Administration

Use the Data Element Library Administration facility for migration, promotion, and updating capabilities.

Option	Description
MD	Migrate data elements and contexts from one data element library to another. If the context does not exist in the destination, it is added to the destination context file.
MS	Migrate screens and their data elements from one Project and Group to another.

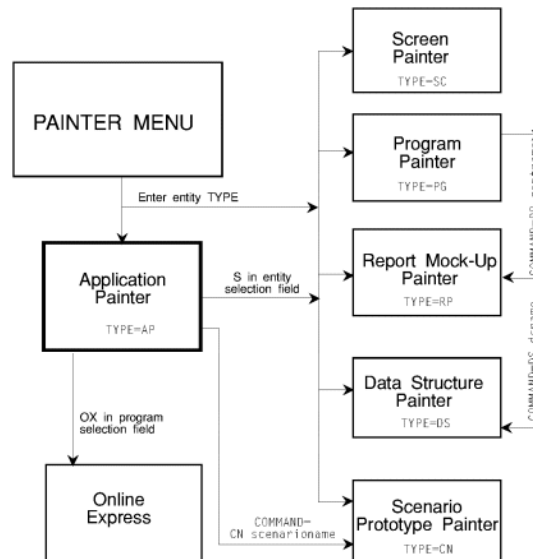
Option	Description
PD	Promote the context of a data element to another context within a data element library.
PS	Promote the context of all the data elements used by a screen to another context. The data elements references in the screen definition are updated to the new context.
SO	Update data element references in the screen definitions to a new context. The data elements with the new context should already exist in the data element library specified in the source.

- 1 If you selected any of the Cross DE Library function options, enter a destination Project and Group.
- 2 Enter a source Project and Group, unless you want to use the environment Project and Group and DE library settings.
- 3 If the Data Element Library location is different from the source or destination Project and Group, specify a DE prefix in the Data Element Library Prefix field.
- 4 Enter the name of the screen in the Screen field or leave this field blank and press Enter to display a selection list.
- 5 Enter the name of the data element in the Element field or leave this field blank and press Enter to display a selection list.
- 6 Enter *S* or *D* in the Collision Priority field if the destination element/context already exists.
- 7 To prevent the Conflict Resolution screen from displaying, enter YES in the Global field.
- 8 To optionally perform this process in batch mode, enter YES in the Batch Execution field.

8 Data Structures

This chapter contains information on navigating using APS painters, coding data structures, and the data structures themselves.

Painter Navigation



Coding Data Structures

Observe the following when coding data structures:

- Code a data structure as follows:

dataelementname PICformat

Dataelementname must be valid COBOL name, maximum 30 characters, or expression introduced by one of the macro default symbols, &, <, \$, or]. PICformat must be a valid COBOL or APS shorthand picture format.

- Do not enter the hierarchical level numbers, except for 66, 77, or 88. Indent each level (except the first) of a data structure a consistent number of spaces.
- Enter one *dataelementname* per line only.
- You cannot use the APS shorthand picture format in conjunction with the PIC clause; you must use valid COBOL syntax.
- Code comments as follows:

```
* comment  
/* comment  
%* comment
```
- Code continuation as follows: ellipsis followed by a space (...).

Data Structures

RENAMES

66 *anytext*

88

```
88     dataname VALUE 'value1' [THRU 'value2']  
...                     'value3' [THRU 'value4']  
...                     'value5' [THRU 'value6']  
...                     .  
...                     .  
...                     'valueN' [THRU 'valueN']
```

Edit Masks

A P Z + - * B \$ 0

OCCURS

(*number*) [TIMES]
(*number1*-*number2*)
dataname (*number1* TO *number2*)
(*&variable*)
(*&variable1* TO *&variable2*)

[... DO|ODO dataname]
[... IX|IB|IXB|IXBY dataname]
[... ASCENDING KEY IS dataname]
[... DESCENDING KEY IS dataname]
... *PICformat*

Picture Formats

APS Shorthand	Generated COBOL
9	PIC 9.
99	PIC 99.
999	PIC 999.
9999	PIC 9999.
9(<i>n</i>)	PIC 9(<i>n</i>).
9 <i>n</i>	PIC 9(<i>n</i>).
9 <i>&variable</i>	PIC 9(<i>&variable</i>).
S <i>&variable</i>	PIC S9(<i>&variable</i>).
SYNC	SYNCHRONIZED
V(<i>n</i>)	PIC V9(<i>n</i>).
V <i>n</i>	PIC V9(<i>n</i>).
V <i>&variable</i>	PIC V9(<i>&variable</i>).
V <i>valueclause</i>	VALUE <i>valueclause</i>
X	PIC X.
XX	PIC XX.

APS Shorthand	Generated COBOL
<i>XXX</i>	<i>PIC XXX.</i>
<i>XXXX</i>	<i>PIC XXXX.</i>
<i>Xn</i>	<i>PIC X(n).</i>
<i>X(n)</i>	<i>PIC X(n).</i>
<i>X&variable</i>	<i>PIC X(&variable).</i>
<i>...</i>	Continuation of a data element.

REDEFINES

dataname REDEF[INES] | R PICformat

VALUE

dataname PICformat
[...] [VALUE|V] 'valueclause'

Commands

Enter any of the following in the Command field.

<i>CONVERT CONV</i>	Convert shorthand format to valid COBOL format to preview results of generated code.
<i>GENERATE GEN [number]</i>	Convert data structure to valid COBOL format and export it to copylib. Number = level number.
<i>UNCONV UNC</i>	Cancel the effects of, or unconvert, the CONVERT command. Required after CONVERT.
<i>VALIDATE VAL</i>	Validate syntax and display error messages.

9 ENDEVOR Interface

This chapter describes the APS Endevor Interface, and includes information on the checkin and checkout of members, revisions, and reports.

Check In a Component

Sign in, add, or update an APS component on the Checkin screen. Optionally add or update the ENDEVOR library with the component.

Field	Description and Values
Entity Type	Entity Type of the APS component to check in. Valid values: <i>AP</i> Application Painter component in APSAPPL plus its related component in APRAPPL <i>CN</i> Scenario Painter component in APSCNIO <i>DS</i> Data Structure Painter component in APSDATA <i>OX</i> Online Express component in APSEXP <i>PG</i> Program Painter component in APSPROG plus its related component in APRPROG <i>RP</i> Report Mock-up Painter component in APSREPT <i>SC</i> Screen Painter component in APSSCRN For other APS component types in your Project.Group, specify a data set name, such as USERMACS and DDISYMB.

Field	Description and Values	
Member	Component name to check in, or leave the Member field blank to select from a member list.	
System	ENDEVOR system name, if different from the default system name for your current APS Project.Group.	
Subsystem	ENDEVOR subsystem name, if different from the default subsystem name for your current APS Project.Group.	
Comment	Text comment for the checkin.	
CCID	ENDEVOR CCID for the checkin.	
Bypass Gen Processor	NO	Default. Do not bypass the associated ENDEVOR Generate Processor
	YES	Bypass the processor.
Delete Input Source	NO	Default. Do not delete the component from the APS Project.Group.
	YES	Delete the component.
Processor Group	Name of ENDEVOR Processor Group.	
Override Signout	NO	Default. Do not override an existing signout.
	YES	Override the signout. You must have authority to do so.
Signin Only	NO	Default. Do not release a previous signout.
	YES	Signin only. Release a previous signout of the component issued with your user ID; the Add or Update action is not executed.
Stage	ENDEVOR Stage number for signin.	
Component Parts	For checking in AP and PG components. Valid values:	
	NONE	Default. Process only the component specified in the Member field.

Field	Description and Values
<i>ALL</i>	Process the component specified in the Member field and all its associated component parts, or components.
<i>LIST</i>	Display the Component Types Selection screen, to select the associated component types for processing.

Check Out a Revision

Retrieve and sign out a revision from a controlled member of the ENDEVOR library for modification on the Checkout screen.

Field	Description and Values
Entity Type	Entity Type of the APS component to check out. Valid values same as for check in.
Member	Member name to checkout, or leave the Member field blank to select from a member list.
System	ENDEVOR system name, if different from the default system name for your current APS Project.Group.
Subsystem	ENDEVOR subsystem name, if different from the default subsystem name for your current APS Project.Group.
Stage	ENDEVOR stage number of the member to check out.
Version	Default is current revision.
Level	Default is current level.
Comment	Text comment for the check out.
CCID	ENDEVOR CCID to associate with the check out.
No Signout	<i>YES</i> Check out and browse the member without signing it out to your user ID. <i>NO</i> Sign out.

Field	Description and Values
Replace Member	<div>YES Overlay an existing member in the APS Project.Group.</div> <div>NO Do not overlay a member.</div>
Override Signout	<div>NO Default. Do not override an existing signout.</div> <div>YES Override the signout. You must have authority to do so.</div>
Component Parts	For checking out AP and PG components. Valid values same as for check in.

Reports

APS/ENDEVOR provides the following reports:

Report	Contents
View Differences	Source statements that differ between a specific component revision and the preceding one. Equates to the ENDEVOR/MVS Change report.
View Print Browse	<div>Log and source change information on:</div> <ul style="list-style-type: none">• All revisions, including creators; creation dates; number of statements; CCIDs; comments; when the component was last generated and retrieved, and by whom.• All statements in the specified revision, marked with the level number at which they were first inserted. <div>Equates to the Browse Element Display report.</div>
View Print History	<div>Log and source change information on:</div> <ul style="list-style-type: none">• All revisions, including creators; creation dates; number of statements; CCIDs; comments; when the component was last generated and retrieved, and by whom.

Report	Contents
	<ul style="list-style-type: none">• All inserted and deleted statements that ever existed in all revisions of the component, marked with the level number at which they were inserted or deleted.
	Equates to the History Element Display report.
View Print Master	Log and source change information on a component, including its processor group; the last action performed against it; its current signout status; when it was last modified and generated, and by whom; the origin of its base revision; who moved or transferred the component from a stage, and when. Equates to the Master Element Display report.
View Print Summary	Log and source change information on all revisions, including creators; creation dates; number of statements; number of inserted and deleted statements. Equates to the Summary Element Display report.

Run the reports from the APS/ENDEVOR Version Control Menu, selecting Option 3 or 4, as desired. The reports default to the current version and level.

10 Generation

This chapter contains information on generating applications, including procedures, parameters, and regeneration.

Generate Procedures

Do the following:

- 1 Select any generation parameters (see following topics).
- 2 Display your application on Application Painter.
- 3 Generate entire application or generate your screens followed by your programs, as follows:

To Generate...	Enter...
Single screen	<i>GENERATE SC screenname</i> in Command field, or <i>G</i> in screen selection field
Multiple screens	<i>G</i> in screen selection fields
All screens	<i>GENERATE SC ALL</i> in Command field
Single program	<i>GENERATE SC programname</i> in Command field, or <i>G</i> in program selection field
Multiple programs	<i>G</i> in program selection fields
All programs	<i>GENERATE SC ALL</i> in Command field
Entire application	<i>GENERATE</i> in Command field
ISPF prototype	<i>ISPF</i> in DC field, <i>GENERATE</i> in Command field

Parameters

Verify or change default parameters that define your development and application target environments. Access from Application Painter by entering *OPT* in the Command field, or select from APS Main Menu.

Reset the following to their original values at installation time:

- | | |
|-------------------|---------------------------|
| BIND options | Job control cards |
| SQL | Library prefixes and DSNs |
| Generator options | APS precompiler options |
| IDMS options | |

- 1 Transfer to Generator Options to define options for the generation process.
- 2 Transfer to Project Group Environment to define your project and group settings from a centralized location.
- 3 Transfer to Precompiler Options to vary some APS parameters during program precompilation.
- 4 Transfer to Report Options to define characteristics that apply to all your APS documentation reports.
- 5 Transfer to SQL Bind and Translate Options to specify DB2 BIND options.
- 6 Transfer to Job Cards to establish up to five job cards with varying information.
- 7 Transfer to IDMS Options to define IDMS-specific parameters.

Generation Options

- 1 Use APS Generator Options to define options for the generation process.

Option	Description and Values	
Target	Define operating system (OS), database (DB), data communications (DC) and SQL targets.	
Job Class	Any job class valid at your site and known to the APS generators.	
Job Dest	Site-specific.	
MSG Class	Site-specific.	
LISTGEN	YES	Provide a listing of generated code.
	NO	Default. Do not provide listing.
COBOL	YES	Save generated COBOL program source in the library or directory that is appropriate for your DC target.
	NO	Default. Do not save COBOL source.
Object	YES	Save generated object code in appropriate library.
	NO	Default. Do not save object code.
MFS/BMS	YES	Save generated MFS or BMS in the library or directory appropriate for your IMS or CICS target.
	NO	Default. Do not save.
GENSRC	YES	Save generated source code in the GENSRC directory or data set.
	NO	Default. Do not save GENSRC.
APS Parm	Override the APS Parm field on the Precompiler Options screen.	
COBOL Parm	Specify parameters or directives for COBOL compiler.	
CARDIN Member	Specify the CNTL library APSDBDC member.	
Generate COBOL II	YES	Generate COBOL II code.
	NO	Do not generate.

Option	Description and Values	
Generate COBOL II	YES	Compile with VS-COBOL II. Use VS-COBOL II directive for Micro Focus COBOL.
	NO	Default. Compile with standard COBOL.
CICS Release	CICS release at your site.	
IMS Release	IMS release at your site.	
SUPRA	YES	Pass native SUPRA procedural statements through APS unchanged.
	NO	Process SUPRA procedural statements.

Project Group Environment

Define your project and group settings on APS Project Group Environment.

Option	Description and Values
Project	Project name. Maximum 8 alphanumeric characters; first character alphabetic.
Group	Group name. Maximum 8 alphanumeric characters; first character alphabetic.
DDIFILE	Location of data set or directory where APS stores extracted information for DBD and DDI statements. Do not specify the name DDIFILE. Default is current project and group.
Data Element Library Prefix	Location of the Data Element Facility APSDE directory or data set; do not specify the name APSDE.

Precompiler Options

Select various APS parameters to obtain special features during program recompilation on APS Precompiler Options. The APS Parm field displays the modified parameters.

The following lists the parameters and their valid values. The Keywords columns specifies the keywords generated by the parameter; you can enter these keywords in the APS Parm field on the Generator Options screen.

Option	Description and Values	
APOST		Overrides QUOTE.
	YES	Default. Lets you use the apostrophe character to delimit non-numeric literals in your input source.
QUOTE		Overrides APOST.
	YES	Lets you use the single quote character to delimit non-numeric literals in your input source.
	NO	Default.
SCBTRACE	YES	Activates the SAGE-TRACE-FLAG debugging facility.
RWT	YES	Default. Generate COBOL code from APS Report Writer statements. Specify with COBOL II compiler.
	NO	Pass Report Writer statements directly to the COBOL compiler.
		For very large programs, enter <i>rwt=bigrwt</i> in the APS Parm field on the Generator Options screen.
MOCKUP-FMP	YES	Scan lines in report mock-ups and processes the characters % \$ & and + as Customization Facility symbols.
	NO	Default.
SUBR	YES	Generated source is a subroutine program.
	NO	Default. Generated source is a primary program.
NARROW	YES	Default. 80 columns is the message report width.
	NO	132 columns is the width.
EVALMESS	YES	Generate messages that list evaluation bracket resolutions. Usually results in long listings.

Option	Description and Values	
GENSEQ	<i>NO</i>	Default.
	Overrides SPACESEQ.	
	<i>YES</i>	Default. Generate sequence numbers in columns 1-6 for blank or out-of-sequence lines of source code and when new lines are generated.
	Overrides GENSEQ.	
SPACESEQ	<i>YES</i>	Generate spaces in columns 1-6; incompatible with LANG=TEXT.
	Overrides GENSEQ.	
SPACEIDENT	See also, GENIDENT, IDENT, SEQ.	
	<i>YES</i>	Generate spaces in columns 73-80. Incompatible with LANG=TEXT.
FMP	<i>YES</i>	Default. Process APS macros and user-defined Customization Facility macros.
	<i>NO</i>	Use only with your own JCL skeleton.
Source	<i>YES</i>	Print the main input source program, specified in the MAIN option, after the message report.
	<i>NO</i>	Default.
GENDIRECT	<i>YES</i>	Allow generation of nested IF statements in the COBOL source.
	Location of the main input source.	
MAIN	<i>MAININ</i>	=YES
	<i>INSTREAM</i>	=YES
	Member Name = <i>membername</i>	
	Specify which processing step(s) that APS performs. You can stop processing at any of the steps listed below. All options except ALL are mutually exclusive.	
	<i>ALL</i>	=YES
	<i>SCB</i>	=YES
XLATE	<i>FMP</i>	=YES
	<i>RED</i>	=YES
	<i>RWT</i>	=YES

Option	Description and Values
EMARK	Three-character string that marks error and warning lines of the message report. Questions= <i>YES</i> Dollars= <i>YES</i> 3-Char String= <i>string</i>
IDENT	See also, GENIDENT, SPACEIDENT, SEQ. <i>YES</i> Generate the internal program name in columns 73-80. <i>NO</i> Default.

SQL BIND Options

The APS Bind Options screen let you specify parameters and change default values for your SQL database target.

Field	Description and Values
Database	Database name. Required for OS/2 Database Manager.
DB2 System Name	Appropriate name for your site.
Plan Name	Plan name you use when you Bind an application. If you leave this field blank, the default depends upon your use of the BIND command in the Application Painter.
Owner of Plan (Authid)	Leave this field blank or specify a primary or secondary authorization ID of the BIND.
Qualifier	Leave this field blank or specify the implicit qualifier for the unqualified table names, views, indexes, and aliases contained in the plan.
Action	Specify the bind action to be executed. Valid values: <i>ADD</i> or <i>REPLACE</i> .
Retain Execution Authority	Used only when you specify <i>REPLACE</i> in the Action field. Valid values: <i>YES</i> or <i>NO</i> .
Isolation Level	Valid values: <i>RR</i> or <i>CS</i> .
Plan Validation Time	Valid values: <i>RUN</i> or <i>BIND</i> .

Field	Description and Values
Explain Path Selection	<p><i>YES</i> Activate the DB2 EXPLAIN function.</p> <p><i>NO</i> Do not activate the function.</p>
Resource Acquisition Time	<p>Valid values: Use or Allocate.</p> <p>If you enter <i>ALLOCATE</i>, enter <i>DEALLOCATE</i> in the Resource Release Time field.</p>
Resource Release Time	<p>Valid values: <i>COMMIT</i> or <i>DEALLOCATE</i>.</p>
Defer Prepare	<p><i>YES</i> Generate the keyword DEFER(PREPARE), which defers the prepare statement referring to a remote object.</p> <p><i>NO</i> Default.</p>
Cache Size	<p>Size (in bytes) of the authorization cache to be acquired in the EDMPOOL for the plan. Valid values: 0 to 4096.</p>
Data Currency	<p><i>YES</i> Data currency is required for ambiguous cursors.</p> <p><i>NO</i> Data currency is not required.</p>
Current Server	<p>Leave this field blank or specify a connection to a location before the plan runs.</p>
Message Flag	<p>Specify which messages display. Valid values: <i>I</i>, <i>W</i>, <i>E</i>, <i>C</i>, or blank.</p>

Job Control Cards

Establish up to five job cards with varying job names, account information, classes, and other attributes on Job Control Cards. After these are defined, specify the job card you want (such as J1 or J2) in the Job Class field of other APS screens.

IDMS Options

The APS IDMS DB Options screen let you specify parameters and change default values for your IDMS database target.

Field	Description and Values	
Dictionary Name	Specify the dictionary name.	
Central Version or Local	Compile environment. APS generates a SYSTRNL with a unique DSN whose high level qualifier is your user ID.	
	<i>CV</i>	Default. Central Version.
	<i>LOCAL</i>	Also enter a volume in the IDMS Local Jrnl Disk Vol field.
	<i>DUMMY</i>	APS generates a SYSTRNL DD DUMMY.
IDMS Local Jrnl Disk Vol	Local compile disk volume for journal.	
Dictionary Update	<i>YES</i>	Log program compile information to the dictionary.
	<i>NO</i>	Default. Do not log.
IDMS DMLC Output to PDS	<i>YES</i>	Write DMLC compile statements to a PDS. Allocate a &DSN..IDMSOUT PDS prior to compilation.
	<i>NO</i>	Default. Do not write DMLC compile statements.
IDMS Loadlib Qualifier	Specify full qualifiers for IDMS..LOADLIB.	
IDMS SYSCTL DSN	Specify DSN of IDMS dictionary.	
CV Node Name	Name of central version DDS node under which LOADLIB program is compiled.	
DMLIST (List Generation)	<i>YES</i>	Generate list.
	<i>NO</i>	Default. Do not generate list.
Generate DB-BIND in Pgm	<i>YES</i>	Do not suppress the generation of the DB-BIND macro.

Field	Description and Values
	<i>NO</i> Suppress the generation of the DB-BIND macro. Code the DB-BIND macro in your program.
IDMS Password	N/A

Regenerate Procedures

Follow the procedures below when you make a change to your application.

Add

To add a screen:

- 1 Modify the application definition on Application Painter.
- 2 Modify the program(s) that utilize the screen in Online Express or on Program Painter.
- 3 Generate the screen.
- 4 Generate and compile the affected program(s).

To add a program, data structure, report mock-up, macro PSB, or subchema:

- 5 Modify the application definition on Application Painter.
- 6 Modify any affected program(s) in Online Express or on Program Painter.
- 7 Generate and compile new or affected program(s).

Modify

To modify DB/DC targets:

- 1 Modify the application definition on Application Painter.

- 2 Modify any other affected entities.
- 3 Generate and compile entire application.
To modify a program, data structure, PSB, or subschema:
- 4 Generate and compile affected programs.
To modify a screen:
- 5 If you modify headings, text, or attributes, generate the screen.
- 6 If you add, delete, or reshuffle fields, generate the screen and then compile all programs using the screen
To modify a macro or report mock-up:
- 7 Compile the program(s) using the entity.

Delete

To delete a screen, program, data structure, report mock-up, macro, PSB, or subschema:

- 1 Modify the application definition on Application Painter.
- 2 Modify any other affected entities.
- 3 Generate and compile entire application.

Rename

To rename an application:

No modification or generation is required.

To rename a program:

- 1 Modify the application definition on Application Painter.
- 2 Modify associated screen(s) parameters on Screen Painter Screen Generation Parameters.
- 3 Generate affected screen(s).
- 4 Generate and compile renamed program.

To rename a screen:

- 5 Modify the application definition on Application Painter.
- 6 Modify the program(s) that utilize the screen in Online Express or on Program Painter.
- 7 Generate the screen.
- 8 Generate and compile the affected program(s).

To rename a data structure, report mock-up, macro PSB, or subschema:

- 9 Modify the application definition on Application Painter.
 - 10 Modify any affected program(s) in Online Express or on Program Painter.
 - 11 Generate and compile new or affected program(s).
-

11 Importers

APS/IMS Syntax

Use DDI statement keyword parameters to specify which database values to import.

DDI Statement	Parameter	Value
*DDI DBD	NAME	<i>Dbdname</i> . Maximum 8 characters.
*DDI REC	NAME	<i>Copylibrecord</i> . Maximum 30 characters. Default is DBD value of <i>SEGM NAME</i> .
	SEG	DBD value of <i>SEG NAME</i> . Maximum 8 characters.
	COPY	<i>Copylibmembername</i> . Optional; maximum 8 characters. Default is <i>SEGM NAME</i> .
	GEN01	Y (default) or N. Optional.
*DDI FLD	NAME	<i>Copylibfieldname</i> . Maximum 30 characters. Default is DBD value of <i>FIELD NAME</i> or DBD value of <i>XDFLD NAME</i> .
	IMSNAME	DBD value of <i>FIELD NAME</i> or DBD value of <i>XDFLD NAME</i> . Maximum 8 characters.
	PIC	<i>COPYLIB PIC</i> . Optional; maximum 24 characters. Default is <i>X(n)</i> , where <i>n</i> = the value of the <i>BYTES</i> keyword in the DBD.

APS/VSAM Syntax

Use DDI statement keyword parameters to specify the VSAM file information to import.

DDI Statement	Parameter	Value
*DDI VSM	DDN	File external ddname. Maximum 8 characters.
	TYPE	K(eyed)-default, E(ntry), or R(elative).
	VSPREFIX	File prefix.
	CYL, TRK, or REC	Choose one.
	CISZ	Control Interval Size. Default is 4096.
	VOL	Volume name.
	CAT	Catalog name.
	Other IDCAMS keywords	IDCAMS syntax; optional.
*DDI REC	NAME	<i>Recordname</i> . Maximum 30 characters.
	SHORT	<i>Shortrecordname</i> . Maximum 8 characters.
	COPY	<i>Membername</i> . Maximum 8 characters.
	SOURCE	Method used to put copylib members in program: <div> <div>P</div> <div>Via an APS %INCLUDE statement.</div> </div> <div> <div>C</div> <div>Via COBOL COPY command.</div> </div>
	MAXLEN	Maximum record length.
	AVGLEN	Average record length. Default is MAXLEN.
IDX	NAME	Copylib record key field name. Maximum 30 characters.

DDI Statement	Parameter	Value
	ALIAS	File that uses multiple copylib records. The name of the field that redefines NAME. Optional; maximum 30 characters.
	TYPE	Index type: <i>P</i> (rimary) First index must be <i>P</i> . <i>U</i> (nique). <i>D</i> (uplicate).
	KEYLEN	Length of key field. Maximum 4 characters.
	OFFSET	The offset position of the field relative to the beginning of the record the first position is 0. Maximum 4 characters.
	DDN	<i>Ddname</i> of the index. Maximum 24 characters. Default is primary <i>ddname</i> .
	PIC	<i>Ddname</i> of the index. Maximum 24 characters. Default is alphanumeric.
	IDCAMS overrides	IDCAMS syntax; optional.
*DDI SUB	NAME	Unique subschema name. Maximum 8 characters.
	RECORD	Value of the first (or only) DDI REC statement NAME keyword. Maximum 30 characters.
	PROCOPT	Optional; process control options: <i>A</i> (ll) - default, <i>G</i> (et), <i>I</i> (nsert), <i>D</i> (elete), or <i>R</i> (eplace)
	ACCESS or ACC	Batch access options; optional. Default is dynamic for KSDS files; sequential for ESDS/RRDS
	BLOCK	0; optional.
	LABEL	<i>STANDARD</i> ; optional.

DDI Statement	Parameter	Value
	ASSIGN or AS	VSAM file external <i>ddname</i> . Optional; maximum 8 characters. Default is value of the DDN parameter in the DDI VSM statement.

12 Keywords

This chapter contains information on using keywords including their placement in code, the structures themselves, and programming commands.

Placement in Code

The following samples include all APS keywords that you can enter in the Program Painter or Specification Editor when creating online or batch programs. They illustrate:

- The program locations where each keyword places the source code after you generate the program through APS to produce an executable COBOL or COBOL/2 program.
- The program locations at which APS places externally-defined components associated with your program, such as user-defined macros and data structures that you list on the Application Painter.

Online programs

```
*SYM1 keyword places Customization Facility code here
*user macros from Appl. Painter, Location = T
*SYM2 keyword places Customization Facility code here
```

```
IDENTIFICATION DIVISION.
```

```
.
.
.
```

```
*/ * keyword places comments here (COBOL/2)
```

```
REMARKS. REM keyword places comments here (COBOL)
```

```
ENVIRONMENT DIVISION.
```

```
.
.
.
```

```

SPECIAL-NAMES.  SPNM keyword places code here
*SYEN keyword places Customization Facility code here

DATA DIVISION.
*SYDD keyword places Customization Facility code here

WORKING-STORAGE SECTION.
*user macros      from Appl. Painter, Location = WT
*data structures  from Appl. Painter, Location = WT
*user macros      from Appl. Painter, Location = WS
*SYWS keyword places Customization Facility code here

*
WS keyword, followed by any of the following six
*   keywords that you enter on the next line,
*   places code here:
*01 keyword
*REC keyword
*DS keyword
*SQL keyword
*++ keyword
*FRFM keyword

*user macros from Appl. Painter, Location = WB
*CA, CA05, CADS keywords place code here in CICS, DDS, IMS,
*   ISPF Dialog, and PM parent pgms

LINKAGE SECTION.
*user macros      from Appl. Painter, Location = LT
*data structures  from Appl. Painter, Location = LK
*user macros      from Appl. Painter, Location = LK
*SYLT keyword places Customization Facility code here
*SYLK keyword places Customization Facility code here

*LK keyword, followed by any of the following six
*   keywords that you enter on the next line,
*   places code here:
*01 keyword
*DS keyword
*REC keyword
*SQL keyword
*++ keyword
*FRFM keyword

*CA, CA05, CADS keywords place code here in ISPF
    and PM child pgms
*user macros      from Appl. Painter, Location = LB

```

```

PROCEDURE DIVISION.  NTRY|PROC keyword generates this stmt
*OPT keyword places code here
.
.
*PARA keyword places code here
.
.
*STUB keyword places code here
.
.
*user macros from Appl. Painter, Location = B
*SYBT keyword places Customization Facility code here
*End of program.

```

Batch programs

```

*SYM1 keyword places Customization Facility code here
*user macros from Appl. Painter, Location field = T
*SYM2 keyword places Customization Facility code here

IDENTIFICATION DIVISION.
.
.
REMARKS.  REM keyword places comments here

ENVIRONMENT DIVISION.
.
.
SPECIAL-NAMES.  SPNM keyword places code here
*SYEN keyword places Customization Facility code here

INPUT-OUTPUT SECTION.
*user macros from Appl. Painter, Location field = IO
*SYIO keyword places Customization Facility code here

FILE-CONTROL.
*IO    keyword places code here
.
.
DATA DIVISION.
*SYDD keyword places Customization Facility code here

FILE SECTION.
*user macros from Appl. Painter, Location field = FD
*FD keyword places code here
*SD keyword places code here

```

```

WORKING-STORAGE SECTION.
*user macros      from Appl. Painter, Location field = WT
*data structures  from Appl. Painter, Location field = WT
*user macros      from Appl. Painter, Location field = WS
*SYWS keyword places Customization Facility code here

*WS keyword, followed by any of the following six
* keywords that you enter on the next line,
* places code here:
*01 keyword
*REC keyword
*DS keyword
*SQL keyword
*++ keyword
*FRFM keyword

*user macros from Appl. Painter, Location field = WB

LINKAGE SECTION.
*user macros      from Appl. Painter, Location field = LT
*data structures  from Appl. Painter, Location field = LK
*user macros      from Appl. Painter, Location field = LK
*SYLT keyword places Customization Facility code here
*SYLK keyword places Customization Facility code here

*LK keyword, followed by any of the following six
* keywords that you enter on the next line,
* places code here:
*01 keyword
*DS keyword
*REC keyword
*SQL keyword
*++ keyword
*FRFM keyword

*user macros      from Appl. Painter, Location field = LB

PROCEDURE DIVISION.          /*NTRY keyword generates this stmt
PROCEDURE DIVISION USING... /*PROC keyword generates this stmt

DECLARATIVES.                /* DECL keyword generates this stmt
DECLARATIVES SECTION. /* DPAR generates this stmt
*DPAR keyword places code here
END DECLARATIVES.

*OPT keyword places code here
.
```



```
.
.
*PARA keyword places code here
.
.
.
*STUB keyword places code here
.
.
.

*user macros from Appl. Painter, Location field = B
*SYBT keyword places Customization Facility code here
*End of program.
```

Structures

++

```
-KYWD- 12-*----20---*----30---*----40---*----
++      PANVALETmembername
```

01

Format 1, define input or output files:

```
-KYWD- 12-*----20---*----30---*----40---*----50---*----60
IO      filename ASSIGN [TO] ...
        ORGANIZATION IS ...
01      input|outputrecordname          PIC clause
```

Format 2, define a data structure:

```
-KYWD- 12-*----20---*----30---*----40---*----50---*----60
01      COBOLdatastructure
        [05 COBOLdatastructure]
```

Format 3, copy a data structure:

```
-KYWD- 12-*----20---*----30---*----40---*----50---*----60
01      COBOLcopystatement
```

Format 4, define Report Writer line types:

```
-KYWD- 12-*---20---*---30---*---40---*---
01      [dataname] TYPE [IS] reportgroup
        Report Writer statements
```

CA

Format 1:

```
-KYWD- 12-*---20---*---30---*---40---*---
CA      datastructure
```

Format 2

```
-KYWD- 12-*---20---*---30---*---40---*---
CA05    COBOLdatastructure
```

Format 3:

```
-KYWD- 12-*---20---*---30---*---40---*---
CADS    datastructurename
```

DECL

```
-KYWD- 12-*---20---*---30---*---40---*---
DECL    declarativestements
```

DPAR

```
-KYWD- 12-*---20---*---30---*---40---*---
DPAR    sectionname SECTION
        USE declarativesentence
[DPAR    paragraphname
        paragraphstatements ]
```

DS

```
-KYWD- 12-*---20---*---30---*---40---*---
DS[nn]  datastructurename
```

FD

Format 1:

```
-KYWD- 12-*----20---*----30---*----40---*----
FD      filename
        [filedescriptionstatement]
```

Format 2:

```
-KYWD- 12-*----20---*----30---*----40---*----
FD      filename
        [filedescriptionstatement]
        REPORT IS|REPORTS ARE report1 [... report15]
```

FRFM

Format 1, for Working-Storage and Linkage Sections only:

```
-KYWD- 12-*----20---*----30---*----40---*----50---*----60
FRFM    COBOLstatements|S-COBOLstatements
```

Format 2, for Procedure Division only:

```
-KYWD- 12-*----20---*----30---*----40---*----50---*----60
        ENTER COBOL|ENTER S-COBOL|++INCLUDE membername
        COBOLstatements|S-COBOLstatements
```

IDCS

```
-KYWD- 12-*----20---*----30---*----40---*----
IDCS    IDMSControlSectionstatements
```

IDSS

```
-KYWD- 12-*----20---*----30---*----40---*----
IDSS    IDSSSchemaSectionstatements
```

IO

Format 1:

```

-KYWD- 12-*---20---*---30---*---40---*---
IO      COBOLselectstatement
        IOstatement

```

Format 2:

```

-KYWD- 12-*---20---*---30---*---40---*---
IO      ASSIGN [TO] COBOLdataname
        "literal"
        EXTERNAL externalfile
        ORGANIZATION IS LINE|RECORD SEQUENTIAL

```

LK

```

-KYWD- 12-*---20---*---30---*---40---*---
LK
kywd    associated data structure

```

MOCK

```

-KYWD- 12-*---20---*---30---*---40---*---
MOCK    mockupname

```

NTRY

See Data Communication Calls.

OPT

```

-KYWD- 12-*---20---*---30---*---40---*---
OPT     PROG

```

PARA

Format 1:

```
-KYWD- 12-*---20---*---30---*---40---*---
      PARA paragraphname [SECTION]
           paragraphcode
```

Format 2:

```
-KYWD- 12-*---20---*---30---*---40---*---
      PARA paragraphname(argument1,argument2,
           ... argument3, ... argumentN)
           paragraphcode
```

PROC

```
-KYWD- 12-*---20---*---30---*---40---*---
      PROC [variablename1 variablename2 ... variablenameN]
```

REC

```
-KYWD- 12-*---20---*---30---*---40---*---
      REC datastructure
```

RED

```
-KYWD- 12-*---20---*---30---*---40---*---
      RED reportname
```

REM

```
-KYWD- 12-*---20---*---30---*---40---*---
      REM commentline1
      .
      .
      .
      REM commentlineN
```

SD

```
-KYWD- 12-*---20---*---30---*---40---*---
SD      sortfilename
        [sortdescription]
```

SPNM

```
-KYWD- 12-*---20---*---30---*---40---*---
SPNM    statement
```

SQL

```
-KYWD- 12-*---20---*---30---*---40---*---
SQL     SQLdatastructurestatement
```

STUB

```
-KYWD- 12-*---20---*---30---*---40---*---
STUB    stubname
```

SY Macro Keywords

```
-KYWD- 12-*---20---*---30---*---40---*---
SYBT    macrocode
SYEN    macrocode
SYDD    macrocode
SYFD    macrocode
SYIO    macrocode
SYLK    macrocode
SYM1    macrocode
SYM2    macrocode
SYRP    macrocode
SYWS    macrocode
```

WS

```
-KYWD- 12-*-----20---*-----30---*-----40---*-----  
WS  
kywd  associated data structures
```

Programming Commands

Enter the following commands in the Command field on Program Painter or Specification Painter.

Command	Description
<i>CONVERT CONV</i>	Convert code to preview results of generation.
<i>DATA DS datastructurename</i>	Transfer to named data structure in Data Structure Painter. END returns to invoking painter.
<i>REPT RP reportname</i>	Transfer to Report Mock-Up Painter and display specified mock-up. END returns to invoking painter.
<i>UNCONV UNC</i>	Cancel the effects of, or unconvert, the CONVERT command. Required after CONVERT.
<i>VALIDATE VAL</i>	Validate syntax and display error messages.

13 Limits

APS enforces the following size and programming limitations.

COBOL/2

Item	Max
Characters in paragraph name	24

Customization Facility

Item	Max
Indents	50
Nested macros	139
Macro call arguments	1000
Nested INCLUDEs	10
DECLARE statements:	
Subscripts	300
Length of subscript	12
Tables	200
Parts per table	1000
Length of a table part	78
System limits:	
Work files (beginning with WORK4)	8
LRECL for INCLUDE lib	80
LRECL for extended INCLUDE lib	140

Online Express

Item	Max
Database access specifications	50
Record name occurrences	20
Field qualification occurrences	70
Scrolling for repeated blocks	1
Field mapping for repeated blocks	2
Subschema limits:	
Files, databases, tables	130
DB2 tables	99
Records	160
Qualifiable fields	990

Painters

Item	Max
Application Painter:	
Associated screens	30
Associated data structures	60
Associated USERMACs	90
Screen Painter:	
I/O and text fields	500
1-byte fields in ISPF	25
Characters in field name	16
Field attributes per ISPF screen	127
Trancode construction fields	8
Scenario Painter:	
Fields per screen	400

Report Writer

Item	Max
Report mock-up lines:	
Coded in Copylib	200
Coded in Report Mock-Up Painter	200
Reports for FD	15
SOURCE/SUM/ VALUE statements	300

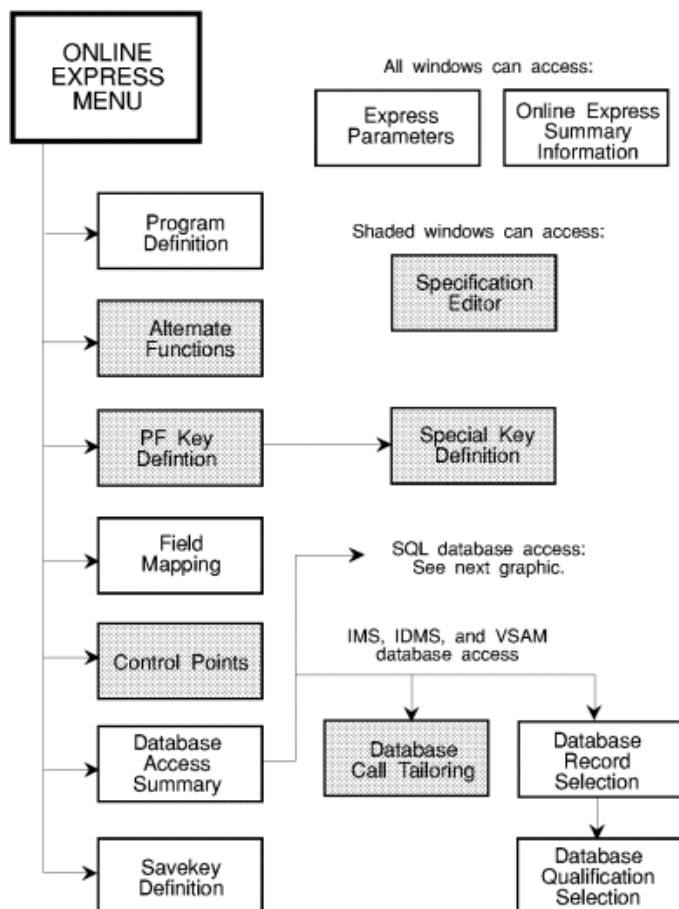
S-COBOL

Item	Max
Paragraph	600
Characters in paragraph name	24
Indentation levels per nested IF structure	14
Paragraph arguments per program	400
EVALUATE statement:	
Conditional fields	255
WHEN conditions	102
Symbol table entries	1801

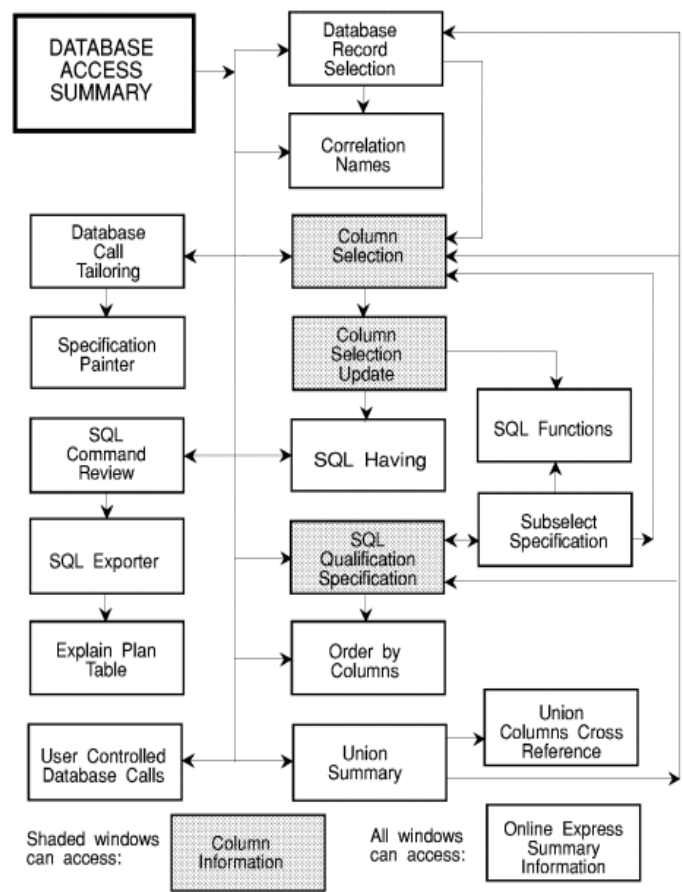
14 Online Express

This chapter contains information on navigating in Online Express, SQL database access, screens, and control points.

Screen Navigation



SQL Database Access Screen Navigation



Screens

Alternate Functions

Define teleprocessing and user-defined functions for character programs; this screen initially displays the function codes entered on the Program Definition screen.

- Access from: Online Express Menu or any primary screen, enter 2 in the Command field.
- Transfer to: Previous, or invoking, screen, press F3.
Program Definition screen, enter 1 in the Command field.
PF Key Functions screen, enter 3 in the Command field.
Field Mapping screen, enter 4 in the Command field.
Control Points screen, enter 5 in the Command field.
Database Access screen, enter 6 in the Command field.
Savekey Definition screen, enter 7 in the Command field.
Specification Editor screen, enter E in the Function field.

Field	Description and Values
Program Input	Function codes from the Program Definition screen are automatically displayed. Add your own function code, 8-character maximum.
Function	Processing action. Valid codes are: * Execute Online Express predefined function. \$ Invoke user macro.

Field	Description and Values																
<i>A</i>	Alias. Specifies that the function code in adjacent field is equivalent to the function code in Function Name field.																
<i>C</i>	Call subroutine (LINK in CICS).																
<i>E</i>	Display Specification Editor for writing local stub. <i>E</i> changes to L after local stub is written.																
<i>G</i>	Execute global stub.																
<i>L</i>	Execute local stub.																
<i>M</i>	For IMS or ISPF prototyping. MSG-SW. Transfer control to program.																
<i>P</i>	Perform paragraph.																
<i>S</i>	Send screen.																
<i>X</i>	For CICS, ISPF Dialog, or ISPF prototyping. XCTL. Transfer control to program																
Reserved Function or Function Name	Object of action coded in function field. Valid actions and their objects are: <table><tr><th>Code</th><th>Function/Name</th></tr><tr><td><i>*</i></td><td><i>*ADD ADD-ROW</i> <i>*BACKWARD</i> <i>*CLEAR</i> <i>*DELETE DELETE-ROW</i> <i>*FORWARD</i> <i>*NEXT</i> <i>*QUERY QUERY-ROW</i> <i>*TERM</i> <i>*UPDATE UPDATE-ROW</i></td></tr><tr><td><i>\$</i></td><td><i>macroname [arguments]</i></td></tr><tr><td><i>A</i></td><td><i>functioncode</i></td></tr><tr><td><i>C</i></td><td><i>programname [userparms]</i></td></tr><tr><td><i>E</i></td><td>Invoke Specification Editor screen</td></tr><tr><td><i>G</i></td><td><i>stubname</i></td></tr><tr><td><i>L</i></td><td>Indicates local stub; value <i>PAINTED</i> or <i>NOT PAINTED</i> displays</td></tr></table>	Code	Function/Name	<i>*</i>	<i>*ADD ADD-ROW</i> <i>*BACKWARD</i> <i>*CLEAR</i> <i>*DELETE DELETE-ROW</i> <i>*FORWARD</i> <i>*NEXT</i> <i>*QUERY QUERY-ROW</i> <i>*TERM</i> <i>*UPDATE UPDATE-ROW</i>	<i>\$</i>	<i>macroname [arguments]</i>	<i>A</i>	<i>functioncode</i>	<i>C</i>	<i>programname [userparms]</i>	<i>E</i>	Invoke Specification Editor screen	<i>G</i>	<i>stubname</i>	<i>L</i>	Indicates local stub; value <i>PAINTED</i> or <i>NOT PAINTED</i> displays
Code	Function/Name																
<i>*</i>	<i>*ADD ADD-ROW</i> <i>*BACKWARD</i> <i>*CLEAR</i> <i>*DELETE DELETE-ROW</i> <i>*FORWARD</i> <i>*NEXT</i> <i>*QUERY QUERY-ROW</i> <i>*TERM</i> <i>*UPDATE UPDATE-ROW</i>																
<i>\$</i>	<i>macroname [arguments]</i>																
<i>A</i>	<i>functioncode</i>																
<i>C</i>	<i>programname [userparms]</i>																
<i>E</i>	Invoke Specification Editor screen																
<i>G</i>	<i>stubname</i>																
<i>L</i>	Indicates local stub; value <i>PAINTED</i> or <i>NOT PAINTED</i> displays																

Field	Description and Values
<i>M</i>	<i>programname [screenname] [keywords]</i>
<i>P</i>	<i>paragraphname</i>
<i>S</i>	<i>screenname [keywords]</i>
<i>X</i>	<i>programname [(NONAPS)] [userparms]</i>
Row Input	
<i>Y</i>	Function applies to a repeated block of records.
<i>N</i>	Function applies to non-repeated records.

Column Selection

Display and select from a list of columns for each database access on the Column Selection screen, and then update criteria by applying functions or specifying Into destinations on the Column Selection Update screen. View column information on the Column Information screen.

Access the Column Selection screen from:	Database Access Summary screen, enter C in the selection field. Subselect Specification screen, enter S in both the column and record selection fields. Union Summary Menu screen, enter C in the selection field.
Transfer to:	Column Selection Update, press F3. Column Information, enter ? in the column selection field.
Access Column Selection Update screen from:	Column Selection screen, press F3.
Transfer to:	Previous screen, press F3. Column Information screen, enter ? in the column selection field. SQL Functions, enter F in the selection field.
Access Column Information screen from:	Column Selection screen, enter ? in the column selection field. Column Selection Update, enter ? in the column selection field.

SQL Qualification Specification, enter ? in the column selection field.

Transfer to:

Previous, or invoking, screen, press the Enter key.

Field	Description and Values								
Distinct	Enter YES to eliminate duplicate rows.								
Function	Built-in functions: <table> <tr> <td>*DA *DATE</td><td>Current date</td></tr> <tr> <td>*TI *TIME</td><td>Current time</td></tr> <tr> <td>*TS *TIMESTAMP</td><td>Timestamp</td></tr> <tr> <td>*CURRENT</td><td>Automatically updates column with correct value</td></tr> </table> Or, enter a valid SQL function. Or, enter <i>F</i> in the selection field to select a SQL function from the SQL Functions screen.	*DA *DATE	Current date	*TI *TIME	Current time	*TS *TIMESTAMP	Timestamp	*CURRENT	Automatically updates column with correct value
*DA *DATE	Current date								
*TI *TIME	Current time								
*TS *TIMESTAMP	Timestamp								
*CURRENT	Automatically updates column with correct value								
Column Name	Lists the columns to be acted on when call is executed. The columns are returned into corresponding host variables if the call is a SELECT, or they are updated with the values in the host variables if the call is an UPDATE or STORE.								
COBOL Name	Override default host variable by typing <i>INTO</i> for SELECTs or a literal for UPDATEs or STOREs.								

Control Points

Insert user-defined logic at standard program control points.

Access from:

Online Express Menu or any primary screen, enter 5 in the Command field.

Transfer to:

Previous, or invoking, screen, press F3.
Specification Editor, enter E in the Action field.

Field	Description and Values
Action	Processing action. See below for values.

Field	Description and Values														
Exit Name	Object of Action coded. Valid Actions and their Exit Names are: <table><tr><td>Code</td><td>Exit Name</td></tr><tr><td>*</td><td>*ADD *BACKWARD *CLEAR *DELETE *FORWARD *NEXT *QUERY *TERM *UPDATE</td></tr><tr><td>\$</td><td><i>macroname [arguments]</i></td></tr><tr><td>E</td><td>Invoke Specification Editor screen</td></tr><tr><td>G</td><td><i>stubname</i></td></tr><tr><td>L</td><td>Indicates local stub; value <i>PAINTED</i> or <i>NOT PAINTED</i> displays</td></tr><tr><td>P</td><td><i>paragraphname</i></td></tr></table>	Code	Exit Name	*	*ADD *BACKWARD *CLEAR *DELETE *FORWARD *NEXT *QUERY *TERM *UPDATE	\$	<i>macroname [arguments]</i>	E	Invoke Specification Editor screen	G	<i>stubname</i>	L	Indicates local stub; value <i>PAINTED</i> or <i>NOT PAINTED</i> displays	P	<i>paragraphname</i>
Code	Exit Name														
*	*ADD *BACKWARD *CLEAR *DELETE *FORWARD *NEXT *QUERY *TERM *UPDATE														
\$	<i>macroname [arguments]</i>														
E	Invoke Specification Editor screen														
G	<i>stubname</i>														
L	Indicates local stub; value <i>PAINTED</i> or <i>NOT PAINTED</i> displays														
P	<i>paragraphname</i>														

Correlation Names

Override the Online Express default correlation name when you specify a JOIN on the Correlation Names screen. Choose your own names, with values up to four characters.

- Access from: Database Access Summary screen, enter N in the selection field.
- Transfer to: Database Access Summary screen, press F3.

Database Access Summary

This screen displays a composite picture of the database accesses and can be used as a menu to access other database access specification screens.

- Access from: Online Express Menu or any primary screen, enter 6 in the Command field.
- Transfer to: Previous screen, press F3.
- Any primary screen, enter its numeral in the Command Field.
- Column Selection screen, enter C in the selection field.
- Correlation Names screen, enter N in the selection field.
- Database Call Tailoring screen, enter T in the selection field.
- Database Qualification screen, enter K in the selection field.
- Database Record Selection screen, enter S in the selection field.
- Order By Columns screen, enter O in the selection field.
- SQL Command Review screen, enter V in the selection field.
- SQL Qualification Specification screen, enter Q in the selection field.
- Union Summary Menu screen, enter U in the selection field.
- User-Controlled Database Calls screen, enter *USER in the Command field.

Field	Description and Values
Selection Field	Enter code for line commands or for transferring to another screen:
A	Line command: After
B	Line command: Before
C	Transfer to Column Selection (SQL only)
D	Line command: Delete
I	Line command: Insert

Field	Description and Values	
	<i>M</i>	Line command: Move
	<i>N</i>	Transfer to Correlation Names (SQL only)
	<i>O</i>	Transfer to Order By Columns (SQL only)
	<i>Q</i>	Transfer to Database Qualification or SQL Qualification
	<i>U</i>	Transfer to Union Summary Menu (SQL only)
	<i>V</i>	Transfer to SQL Command Review (SQL only)
Function	<p>Optionally associate the following functions with one of the adjacent Action values:</p> <p><i>*QUERY</i> <i>*UPDATE</i> <i>*ADD</i> <i>*DELETE</i> <i>*USER</i></p>	
Action	<i>O</i>	Obtain
	<i>S</i>	Store
	<i>M</i>	Modify
	<i>L</i>	Loop
	<i>E</i>	Erase
	<i>P</i>	Position (IMS or IDMS)
Qualifier	<i>*BOOLEAN</i>	Record is qualified on two or more fields or columns.
	<i>*CURRENT</i>	Record access is based on current positioning.
	<i>*KEYQUAL</i>	Record is qualified on a single key field or column.
	<i>*NO-QUAL</i>	Record access is not qualified.

Field	Description and Values
Nesting	Indicates whether an access is nested within a loop call. The default nesting level number displays: <div><div>0</div>Not nested; call is executed independently of other calls.</div> <div><div>- 1</div>Call is nested within the preceding level 0 loop.</div> <div><div>--2</div>Call is nested within the preceding level 1 loop.</div>

You may override the nesting level with a lesser value.

Database Call Tailoring

Modify the processing of database calls. Each modifiable call has its own corresponding Call Tailoring screen.

- Access from: Database Access Summary screen, enter T in the selection field.
- Transfer to: Database Access Summary screen, press F3.
Specification Editor, enter E in the Action field.

Field	Description and Values
Action to be Tailored	<div><div>O</div>Obtain</div> <div><div>E</div>Erase</div> <div><div>M</div>Modify</div> <div><div>S</div>Store</div>
Control Points	Access control points displayed for Modify, Store, Erase, and Obtain.
Action	<div><div>\$</div>Invoke user macro</div> <div><div>E</div>Display Specification Editor for writing local stub. E changes to L after local stub is written</div>

Field	Description and Values
	G Execute global stub
	L Specify local stub
	P Perform paragraph
Control Point Name	Object of Action coded. Valid actions and their objects are: Code Name \$ macroname [arguments] E Press Enter to invoke Specification Editor G stubname L Value PAINTED or NOT PAINTED displays P paragraphname
Status Matrix	Override default status codes for any status flag with one of the following: N Normal condition. X Exception condition; access not in error. E Error condition; access in error, message displays, program aborts when Abort field value is Y. Flag Default OK (OK-ON-REC) N END (END-ON-REC) E NTF (NTF-ON-REC) E DUP (DUP-ON-REC) E VIO (VIO-ON-REC) E
Error Message	Defaults to standard Online Express messages. Other valid values are a text error message or user-defined macro name, without delimiters; 44 characters maximum.
Error Message Type	S Default. Standard Online Express error message identifying record and type of access attempted. T Text message entered in Error Message field. M User macro named in Error Message field.

Field	Description and Values	
Abort on Error	Y	Default. Stop processing when Error status code returned.
	N	Do not stop processing.

Database Qualification

Specify the access qualifications of the Obtain of any record.

Access from:	Database Access Summary screen, enter K in the selection field. Database Record Selection screen, enter S in the selection field.
Transfer to:	Previous screen, press F3.

Field	Description and Values	
Selection Field	I	Insert
	D	Delete
	R	Repeat
TY	Type of field. Values displayed:	
	AD	Address field DB-KEY
	CA	IDMS only. CALC key
	KY	Key field
	PR	VSAM primary index
	SQ	IDMS and IMS only. Field upon which records are sequenced
OP	SR	Nonunique field
	= EQ	Equal to
	> GT	Greater than
	< LT	Less than
	>= GE	Greater than or equal to
	<= LE	Less than or equal to
Value	^= NE	Not equal to
	COBOL data name, variable, or literal on which to qualify field.	

Field	Description and Values
Len	Length of field. Qualify on a VSAM partial key by overriding the full length with a partial length.
Bool	To specify Boolean qualification, type AND or OR in the field.

Database Record Selection

Define database accesses on the record selection list presented by the Database Record Selection screen. You can define multiple accesses for a maximum of 50 records. JOINS and UNIONS are supported for SQL.

- Access from: Database Access Summary screen, enter S in the selection field.
- Union Summary Menu, enter S in the selection field.
- Transfer to : Previous screen, press F3.
- Database Qualification screen, enter S in the selection field.

Field	Description and Values														
Action	<p>Specify one or more database accesses for a record. Database accesses correspond to and complete the definition of database processing functions. Specify DB accesses that correspond to DB functions as follows:</p> <table><tr><th>Access</th><th>Function</th></tr><tr><td>O(btain)</td><td>Query, Forward, Next</td></tr><tr><td>M(odify)</td><td>Update</td></tr><tr><td>E(rase)</td><td>Delete</td></tr><tr><td>S(tore)</td><td>Add</td></tr><tr><td>L(oop)</td><td>Use with Obtain to define record loop; also needed with Forward.</td></tr><tr><td>P(osition)</td><td>Use for IMS and IDMS relationship navigation.</td></tr></table>	Access	Function	O(btain)	Query, Forward, Next	M(odify)	Update	E(rase)	Delete	S(tore)	Add	L(oop)	Use with Obtain to define record loop; also needed with Forward.	P(osition)	Use for IMS and IDMS relationship navigation.
Access	Function														
O(btain)	Query, Forward, Next														
M(odify)	Update														
E(rase)	Delete														
S(tore)	Add														
L(oop)	Use with Obtain to define record loop; also needed with Forward.														
P(osition)	Use for IMS and IDMS relationship navigation.														

Express Parameters

Specify generation options.

Access from: Online Express Menu or any primary screen, enter P in the Command field.

Transfer to: Previous screen, press F3.

Field	Description and Values	
Control Point Comments	YES	Include standard control point comments in the generated program.
	NO	Default. Do not include.
Interscreen Moves	YES	Move screen field values from the sending screen to the receiving screen when the SEND or MSG-SW functions execute.
	NO	Default. Do not move.
Database Calls	YES	Default when subschema defined. Generate APS database calls.
	NO	Default when no subschema. Do not generate.
Clear Screen with LOW-VALUES	YES	Default for CICS applications. Clear repeated record blocks with LOW-VALUES.
	NO	Default for IMS, DDS, and ISPF Dialog applications. Clear repeated record blocks with spaces.
Use SYSMSG Field	PM only.	
	YES	Default. Display system messages in the SYSMSG box.
	NO	Display system messages in a PM message box.
Display Union Support Info	YES	Automatically display Express Union Support screen whenever you define or review a UNION.

Field	Description and Values	
Generate to Program Painter	NO	Do not display.
	YES	Default. Write program to APSPROG.
	NO	Save specifications but do not write to APSPROG.

Field Mapping

Map screen or screen fields to program fields; specify the database record fields that correspond to the screen or screen fields.

Access from: Online Express Menu or any primary screen, enter 4 in the Command field.

Transfer to: Previous screen, press F3.
 Program Definition, enter 1 in the Command field.
 Alternate Functions screen, enter 2 in the Command field.
 PF Key Functions screen, enter 3 in the Command field.
 Control Points screen, enter 5 in the Command field.
 Database Access screen, enter 6 in the Command field.
 Savekey Definition screen, enter 7 in the Command field.

Field	Description and Values	
Screen Field	Automatically supplied. An (RB1) appearing after a screen name indicates the field is in a repeated block.	
Selection Field	D	Delete continuation line.
	E	Select field for edit specification.
	I	Insert continuation line.
I/O/B	I	Input. Data moves from screen or screen to database record.
	O	Output. Data moves from database record to screen or screen.

Field	Description and Values
B	Both (default). Data moves in both directions.
Program Field	Database field name that supplies or accepts data from screen or screen field.

Command	Description
*	Asterisk command. Duplicate the names of all screen or screen fields in the Program Field, except function, savekey, and system message fields. Enter B in I/O/B field. <hr/> Note: Command does not overwrite any program field name previously entered. <hr/>
D and I	Line commands. Specify continuation for qualifying a program field.
FE	Access the GUI Painter or the Screen Painter Field Selection screen to display field edits.
L linenumber	Positions a specific line at the top of screen.
<i>PRE[FIX] fieldprefix [beginlinenumber[endlinenumber]]</i>	Prefix program field with named prefix. An * for beginlinenumber or endlinenumber means the first or last line.
<i>RES[ET] [beginlinenumber[endlinenumber]]</i>	Reset screen by blanking out I/O/B and Program Field fields. An * for beginlinenumber or endlinenumber means the first or last line.

Menu Item Functions

Define teleprocessing and user-defined functions for GUI menu controls.

- Access from: Online Express Menu or any primary screen, enter 2.1 in the Command field.
- Transfer to: Previous, or invoking, screen, press F3.
- Program Definition screen, enter 1 in the Command field.
- Pushbutton Functions screen, enter 2.2 in the Command field.
- Field Mapping screen, enter 4 in the Command field.
- Control Points screen, enter 5 in the Command field.
- Database Access screen, enter 6 in the Command field.
- Savekey Definition screen, enter 7 in the Command field.
- Specification Editor screen, enter E in the Function field.

Field	Description and Values
Menu Item	Menu controls automatically display.
Function	Processing action. Valid codes are:
*	Execute Online Express predefined function.
\$	Invoke user macro.
C	Call subroutine (LINK in CICS).
E	Display Specification Editor for writing local stub. E changes to L after local stub is written.
G	Execute global stub.
L	Execute local stub.
P	Perform paragraph.
S	SEND screen.
X	Transfer control to program.

Field	Description and Values
Reserved Function or Function Name	Object of action coded in function field. Valid actions and their objects are:
	Code Function/Name
	<i>* *ADD</i>
	<i> *CLEAR</i>
	<i> *DELETE</i>
	<i> *NEXT</i>
	<i> *QUERY</i>
	<i> *TERM</i>
	<i> *UPDATE</i>
	<i>\$ macroname [arguments]</i>
	<i>C programname [userparms]</i>
	<i>E Invoke Specification Editor screen</i>
	<i>G stubname</i>
	<i>L Indicates local stub; value PAINTED or NOT PAINTED displays</i>
	<i>P paragraphname</i>
	<i>S screenname [keywords]</i>
	<i>X programname [(NONAPS)] [userparms]</i>

Online Express Information Summary

View summary information about your application, including application, program, screen or screen, and subschema names, and update statistics.

- Access from: Any Online Express screen, enter OXINFO in the Command field.
- Transfer to: Previous screen, press F3.

Online Express Menu

Select screens for program specifications.

Option	Description
1	Display Program Definition screen to specify program information and functions.
2	Display Alternate Functions screen to define application and TP functions.
3	Display PF Key Functions screen to assign PF key functions.
4	Display Field Mapping screen to map screen fields to program fields.
5	Display Control Points screen to add application-specific logic.
6	Display Database Access Summary screen when database access specifications exist; display Database Record Selection screen to specify database access.
7	Display Savekey Definition screen to specify savekey storage requirements.
SC	Invoke APS Screen Painter or APS GUI Painter to change a screen field or screen control.
P	Display Express Parm's screen to set generation options and display Union support.

Command	Description
<i>COPY programname</i>	Copy specifications of an existing Online Express program.
<i>FE</i>	Access the GUI Painter or the Screen Painter Field Selection screen.
<i>GEN</i>	Save program specifications, write program to Program Painter, and generate and compile program.
<i>REPORT REP</i>	Submit batch job to generate Program Definition report, which documents all specifications entered on Online Express screens.
<i>SC</i>	Transfer to the Screen Painter or GUI Painter.

Order by Columns

Specify row retrieval order by adding, deleting, or rearranging the order of columns.

- Access from: Database Access Summary screen, enter O in the selection field.
Union Summary Menu screen, enter O in the Command field.
- Transfer to: Previous screen, press F3.

Field	Description and Values
Loop Max	Define maximum number of SQL processing loops; maximum is 99999.
Selection Field	A After I Insert B Before M Move D Delete ? Obtain column information
Column Name	Online Express default is displayed. Add, delete, or type over column name to reorder.
Seq	A Default. Ascending sequence D Descending
Index	Y Unique N Not unique
Commands	Description
LISTCOL	Display the columns.

PF Key Functions

Associate any function (except the Alias function) to PF keys.

- Access from: Online Express Menu or any primary screen, enter 3 in the Command field.

Transfer to: Previous screen, press F3.
Program Definition screen, enter 1 in the Command field.
Alternate Functions screen, enter 2 in the Command field.
Field Mapping screen, enter 4 in the Command field.
Control Points screen, enter 5 in the Command field.
Database Access screen, enter 6 in the Command field.
Savekey Definition screen, enter 7 in the Command field.
Special Key Definition screen, enter SPC in the Command field.
Specification Editor, enter E in the Function field.

Field	Description and Values
Function	Processing action. Valid codes are: * Execute Online Express predefined function. \$ Invoke user macro. A Alias. Specifies that the function code in adjacent field is equivalent to the function code in Function Name field. C Call subroutine (LINK in CICS). E Display Specification Editor for writing local stub. E changes to L after local stub is written. G Execute global stub. L Execute local stub. M For IMS or ISPF prototyping. MSG-SW. Transfer control to program. P Perform paragraph. S Send screen or screen. X For CICS, ISPF Dialog, or ISPF prototyping. XCTL. Transfer control to program

Field	Description and Values																						
Reserved Function or Function Name	Object of action coded in function field. Valid actions and their objects are: <table><tr><th>Code</th><th>Function/Name</th></tr><tr><td>*</td><td>*ADD ADD-ROW *BACKWARD *CLEAR *DELETE DELETE-ROW *FORWARD *NEXT *QUERY QUERY-ROW *TERM *UPDATE UPDATE-ROW</td></tr><tr><td>\$</td><td>macroname [arguments]</td></tr><tr><td>C</td><td>programname [userparms]</td></tr><tr><td>E</td><td>Invoke Specification Editor screen</td></tr><tr><td>G</td><td>stubname</td></tr><tr><td>L</td><td>Indicates local stub; value PAINTED or NOT PAINTED displays</td></tr><tr><td>M</td><td>programname [screenname screenname] [keywords]</td></tr><tr><td>P</td><td>paragraphname</td></tr><tr><td>S</td><td>screenname screenname [keywords]</td></tr><tr><td>X</td><td>programname [(NONAPS)] [userparms]</td></tr></table>	Code	Function/Name	*	*ADD ADD-ROW *BACKWARD *CLEAR *DELETE DELETE-ROW *FORWARD *NEXT *QUERY QUERY-ROW *TERM *UPDATE UPDATE-ROW	\$	macroname [arguments]	C	programname [userparms]	E	Invoke Specification Editor screen	G	stubname	L	Indicates local stub; value PAINTED or NOT PAINTED displays	M	programname [screenname screenname] [keywords]	P	paragraphname	S	screenname screenname [keywords]	X	programname [(NONAPS)] [userparms]
Code	Function/Name																						
*	*ADD ADD-ROW *BACKWARD *CLEAR *DELETE DELETE-ROW *FORWARD *NEXT *QUERY QUERY-ROW *TERM *UPDATE UPDATE-ROW																						
\$	macroname [arguments]																						
C	programname [userparms]																						
E	Invoke Specification Editor screen																						
G	stubname																						
L	Indicates local stub; value PAINTED or NOT PAINTED displays																						
M	programname [screenname screenname] [keywords]																						
P	paragraphname																						
S	screenname screenname [keywords]																						
X	programname [(NONAPS)] [userparms]																						
Commands	Description																						
*	Copy functions defined for PFKEY01 through PFKEY12 to PFKEY13 through PFKEY24.																						

Program Definition

Specify basic program information and functions.

Access from: Online Express Menu or any primary screen, enter 1 in the Command field.

Transfer to: Previous screen, press F3.
Alternate Functions screen, enter 2 in the Command field.
PF Key Functions screen, enter 3 in the Command field.
Field Mapping screen, enter 4 in the Command field.
Control Points screen, enter 5 in the Command field.
Database Access screen, enter 6 in the Command field.
Savekey Definition screen, enter 7 in the Command field.

Field	Description and Values
Function Field	Name of function code field from associated screen or screen.
Function Codes	Specify one or more predefined function codes to be allowed by program. <div><div>Q</div>Query. Retrieve record. <div>U</div>Update record. <div>A</div>Add. Create new record. <div>D</div>Delete record. <div>N</div>Next. Retrieve next sequential record, based on qualification of previous record accessed. <div>R</div>Refresh. Re-read the database when the end user executes a write function on a repeated record block row and re-display the record block. <div>F</div>Forward. Retrieve next block of repeated records. <div>B</div>Backward. Retrieve previous block of repeated records. <div>C</div>Clear. Move spaces to all screen or screen fields. <div>E</div>Exit. Terminate program.</div>
Row Function Field	Field associated with each record of a repeated block. End user enters row function codes in this field.

Field	Description and Values
Row Function Codes	Predefined row function codes to be allowed by program: U Update record A Create new record D Delete record
SYSMSG Field	System message field specified in Screen Painter.

Savekey Definition

Specify savekey storage requirements. This screen displays automatically calculated savekey storage requirements; specify either screen fields or the program Commarea to fulfill the requirements.

Access from: Online Express Menu or any primary screen, enter 7 in the Command field.

Transfer to: Previous screen, press F3.
 Program Definition screen, enter 1 in the Command field.
 Alternate Functions screen, enter 2 in the Command field.
 PF Key Functions screen, enter 3 in the Command field.
 Field Mapping screen, enter 4 in the Command field.
 Control Points screen, enter 5 in the Command field.
 Database Access screen, enter 6 in the Command field.

Special Key Definition

Associate any function (except Alias) to the ENTER, CLEAR, or PA1-3 (CICS only) keys.

Access from: PF Key Functions screen, enter SPC in the Command field.

Transfer to: PF Key Functions screen, press F3.
 Specification Editor, enter E in Option field.

Field	Description and Values														
Option	<p>Processing action. Valid codes are:</p> <ul style="list-style-type: none"> * Execute Online Express predefined function. \$ Invoke user macro. C Call subroutine (LINK in CICS). E Display Specification Editor for writing local stub. E changes to L after local stub is written. G Execute global stub. L Execute local stub. M For IMS or ISPF prototyping. MSG-SW. Transfer control to program. P Perform paragraph. S Send screen or screen. X For CICS, ISPF Dialog, or ISPF prototyping. XCTL. Transfer control to program 														
Reserved Function or Function Name	<p>Object of action coded in function field. Valid actions and their objects are:</p> <table> <tr> <th>Code</th><th>Function/Name</th></tr> <tr> <td>*</td><td> *ADD ADD-ROW *BACKWARD *CLEAR *DELETE DELETE-ROW *FORWARD *NEXT *QUERY QUERY-ROW *TERM *UPDATE UPDATE-ROW </td></tr> <tr> <td>\$</td><td><i>macroname [arguments]</i></td></tr> <tr> <td>C</td><td><i>programname [userparms]</i></td></tr> <tr> <td>E</td><td>Invoke Specification Editor screen</td></tr> <tr> <td>G</td><td><i>stubname</i></td></tr> <tr> <td>L</td><td>Indicates local stub; value PAINTED or NOT PAINTED displays</td></tr> </table>	Code	Function/Name	*	*ADD ADD-ROW *BACKWARD *CLEAR *DELETE DELETE-ROW *FORWARD *NEXT *QUERY QUERY-ROW *TERM *UPDATE UPDATE-ROW	\$	<i>macroname [arguments]</i>	C	<i>programname [userparms]</i>	E	Invoke Specification Editor screen	G	<i>stubname</i>	L	Indicates local stub; value PAINTED or NOT PAINTED displays
Code	Function/Name														
*	*ADD ADD-ROW *BACKWARD *CLEAR *DELETE DELETE-ROW *FORWARD *NEXT *QUERY QUERY-ROW *TERM *UPDATE UPDATE-ROW														
\$	<i>macroname [arguments]</i>														
C	<i>programname [userparms]</i>														
E	Invoke Specification Editor screen														
G	<i>stubname</i>														
L	Indicates local stub; value PAINTED or NOT PAINTED displays														

Field	Description and Values
M	<i>programname [screenname screenname] [keywords]</i>
P	<i>paragraphname</i>
S	<i>screenname screenname [keywords]</i>
X	<i>programname [(NONAPS)] [userparms]</i>

Specification Editor

Write local program stubs and paragraphs for user-defined functions and control point logic.

Access from: Alternate Functions screen, enter E in the Function field.
 PF Key Functions screen, enter E in the Function field.
 Special Key Definition screen, enter E in the Option field.
 Control Points screen, enter E in the Action field.
 Database Call Tailoring screen, enter E in the Action field.

Transfer to: Previous screen, press F3.

To code program stubs, follow these rules:

- Do not code a PERFORM statement or PARA keyword on the first line; Online Express generates them.
- Code the paragraph statements in S-COBOL, COBOL, or COBOL/2 code, beginning in column 12.
- Paragraphs can be coded on subsequent lines; code a PERFORM statement and the PARA keyword (in the Kywd column) for each paragraph.
- Data Division code may follow Procedure Division code.
- Paragraphs performed at multiple control points should be written at one central location, MISC-USER-PARAGRAPHS, shown on the Control Points screen. MISC-USER-PARAGRAPHS is not a control point, but a location to store paragraphs performed at control points. This location provides access to the Specification Editor,

where you write the paragraphs. When writing paragraphs at MISC-USER-PARAGRAPHS, code:

- Program Painter keyword in the Kywd column on the first line
- The PARA keyword for each paragraph

SQL Command Review

View the SQL that will be generated for a SQL call.

- Access from: Database Access Summary screen, enter V in the selection field.
- Transfer to: Database Access Summary screen, press F3.
SQL Exporter screen, enter SAVE in the Command field.

Command	Description
SAVE	Save the SQL and pertinent S-COBOL and access the SQL Exporter screen, where you can access SPUFI or XDBSQL and test execute the call.

SQL Functions

Apply SQL functions against columns. You can choose a function from a list of all SQL functions, to apply against columns in SELECTs or SUBSELECTs.

- Access from: Column Selection Update screen, enter F in the selection field.
Subselect Specification screen, enter S in the function selection field.
- Transfer to: Previous screen, press F3.

To select a function, type S in the selection field preceding that function. You may also supply additional arguments for the CHAR, DECIMAL, and SUBSTR functions.

SQL Having Clause Specification

Create a HAVING clause, which restricts the columns processed by a GROUP BY clause.

Access from: Automatically displays when you specify GROUP BY syntax on the Column Selection Update screen.

Transfer to: The next screen in the call definition, press F3.

Field	Description and Values
Operator	= EQ Equal to > GT Greater than < LT Less than >= GE Greater than or equal to <= LE Less than or equal to ^= NE Not equal to [NOT] BETWEEN [NOT] LIKE [NOT] NULL [NOT] IN <rel oper> ALL SUBSELECT only [NOT] EXISTS SUBSELECT only <rel oper> ANY SUBSELECT only
Value(s)	COBOL data name, variable, column name, or literal. To continue on subsequent a line, enter I on the current line, type OF on the new line in the Operator field, and enter a qualifier in this field.
Bool	AND or OR

SQL Prototype

Save the SQL generated by the call, or access SPUFI or XDBSQL to test execute the call.

Access from: SQL Command Review screen, enter SAVE in the Command field.

Transfer to: SQL Command Review screen, press F3.

Field	Description and Values	
Option	1	Specify the target library in the fields provided.
	2	Display Explain Plan Table screen to view the structure of the SQL call.
	3	Invoke SPUFI to further test the SQL statement.
Target Dataset for Option 1	Information required to save the SQL.	
Replace Like-Named Members	YES	Update existing member by replacing previous version.
	NO	Display message the member exists.
Save Plan Table Entry	YES	Save.
	NO	Do not save.

SQL Qualification Specification

Qualify the database accesses by specifying access qualifications for a record Obtain.

- Access from: Database Access Summary screen, enter Q in the selection field.
Union Summary Menu screen, enter Q in the selection field.
- Transfer to: Column Information screen, enter ? in the selection field.
Database Access Summary screen, press F3.
Subselect Specification screen, enter S in the selection field.

Field	Description and Values	
Selection Field	A	After
	M	Move

Field	Description and Values
	B Before
	R Repeat
	D Delete
	S Transfer to Subselect Specification
	I Insert
	? Transfer to Column Information
Parenthesis Field	Enter a left parenthesis to begin a group of qualification conditions. Enter a right parenthesis in the Boolean field on a subsequent line.
Corr	Table correlation name for joined tables. Default names are A through P. Specify your own names (maximum 4 characters) on the Correlation Names screen.
Operator	= EQ Equal to > GT Greater than < LT Less than >= GE Greater than or equal to <= LE Less than or equal to ^= NE Not equal to [NOT] BETWEEN [NOT] LIKE [NOT] NULL [NOT] IN <rel oper> ALL SUBSELECT only [NOT] EXISTS SUBSELECT only <rel oper> ANY SUBSELECT only
Value(s)	COBOL data name, variable, SQL column name, literal, or SUBSELECT on which to qualify field.
Boolean	Valid on two or more columns. Specify AND or OR in the field.

Subselect Specification

Specify a subselect clause for a column.

Note: Make sure that the column and record are specified in order to display the column list and to ensure that the SQL function you select is compatible with the data type of the column you select.

- Access from: SQL Qualification Specification screen, enter S in the selection field or SUBSELECT in Value(s) field.
- Transfer to: Column Selection screen, enter S in the column selection field.
- SQL Functions screen, enter S in the function selection field.
- SQL Qualification Specification screen, press F3.

Field	Description and Values
Function	Type a SQL function, or leave this field blank and enter S in the selection field to access the SQL Functions screen, where you can select the desired function.
Column	Type the column name, or leave blank and enter S in the selection field to access the Column Selection screen, where you can select the desired column. A ? in the selection field displays the Column Information screen with information about the column named in this field.
From Record	Enter S next to the record from which to specify a SUBSELECT.
Where Qualification	Specify access qualification based on the column entered and the condition outlined by the Oper and Value fields.
Column	Type the column name, or leave blank and enter S in the selection field to access the Column Selection screen where you can select the desired column.

Field	Description and Values
Oper	= EQ Equal to [NOT] BETWEEN > GT Greater than [NOT] LIKE < LT Less than [NOT] NULL >= GE Greater than or equal to [NOT] IN <= LE Less than or equal to ^= NE Not equal to
Value	Specify value, if appropriate.

Note these general rules about SUBSELECTS:

- Specify ALL columns only when the SUBSELECT forms an EXISTS clause.
- JOINS are not currently supported.
- Nested SUBSELECTS are not currently supported.

Union Summary Menu

The Union Summary Menu leads you through the definition and summary of any UNION clauses you specify in you application. This screen summarizes the SELECT statements that form UNION clauses you specify.

- Access from: Database Access Summary screen, enter U in the selection field.
- Transfer to: Database Access Summary screen, press F3.
- Database Record Selection screen, enter S in the selection field.
- SQL Qualification Specification screen, enter Q in the selection field.
- Column Selection screen, enter C in the selection field.
- Order By Columns screen, enter O in the Command field.

Fields	Description and Values	
Union or Union All selection field	UNION	Select and qualify the rows to be retrieved.
	A	Retrieve all rows from the SELECT result tables
	A	After
	C	Transfer to Column Selection
	B	Before
	Q	Transfer to Qualification Specification
	I	Insert
	S	Transfer to Database Record Selection
Literal Id	M	Move
	Identify which UNION statement was matched in retrieving a row. Enter a literal constant; 8-character maximum.	

If you specify a mismatch of column data types or lengths, the Union Columns Cross Reference screen displays. To correct the mismatch, transfer to Column Selection Update (enter C in the selection field of Union Summary Menu) and make the correction.

Observe the following general rules for UNIONS:

- Data returns in the host variable(s) of the first statement.
- APS uses first SELECT statement as the MAIN statement.
 - If you select an uneven number of columns, the MAIN statement must contain the largest number of columns. If the first SELECT has more columns than subsequent SELECTs, APS pads the subsequent statements with literal strings or numeric values.
 - APS validates the MAIN statement columns against columns from subsequent statements.
- Online Express prompts you for each query step. To view or change the criteria, use the Line Commands in the selection field.

User Controlled Database Calls

You can assign database accesses to a function and execute it within custom code.

Access from: Database Access Summary screen, enter *USER in the call Function field, and USER in the Command field.

Transfer to: Database Access Summary screen, press F3.

Assign database accesses to function *USER and execute it as follows:

- 1 On the Database Access Summary screen, enter *USER in the Function field next to the desired call.
- 2 Enter *USER in the Command field. If desired, override the paragraph name on this screen.
- 3 Perform this paragraph.

Control Points

The control points that apply to your program vary from program to program, depending on which functions you define for the program. To execute logic at control points, you display the Control Points screen, which displays all control points that exist in your program. The complete list of control points is as follows:

Control Point	Location in Program
AFTER-RECEIVE-PARA	After entering a program, regardless of invocation mode.
POST-SCREEN-READ	After a screen-invoked program receives its screen.
TRANSID-INVOKED-PARA	After a transid-invoked program is invoked.
PROGRAM-INVOKED-PARA	When APS displays the screen of a program invoked by the XCTL or MSG-SW function.
PRE-TERM	Before APS terminates the program.

Control Point	Location in Program
AFTER-ENTER-CHECK	After the end user presses the processing key (the Enter key is the default), and before the PRE-FUNCTION-TEST paragraph executes.
PRE-FUNCTION-TEST	Before APS evaluates all functions except the Terminate, or Exit, function.
PRE-BRANCH	Before each MSG-SW, XCTL, or Call function executes.
ED-ERROR-PRE-SEND	Before APS send a screen whose field edits have failed.
GENERAL-PRE-SEND	After APS checks all functions, and before the TP-SEND call executes, when invocation mode is screen-invoked.
BEFORE-SEND-PARA	Before APS sends the screen, regardless of invocation mode.
PRE-SCREEN-TO-REC	Before APS performs the MOVE-SCREEN-TO-REC paragraph.
POST-SCREEN-TO-REC	After APS performs the MOVE-SCREEN-TO-REC paragraph, and the Update or Add function executes.
PRE-REC-TO-SCREEN	Before APS performs the MOVE-REC-TO-SCREEN paragraph.
POST-REC-TO-SCREEN	After APS performs the MOVE-REC-TO-SCREEN paragraph, and after the Query function executes.
PRE-RB1-ROW-TO-REC	Before the Add or Update function executes for a repeated record block row, and before screen fields move to database fields. APS uses the subscript CTR to reference repeated block rows.
POST-RB1-ROW-TO-REC	Before the Add or Update function executes for a repeated record block row, and after screen fields move to database fields. APS uses the subscript CTR to reference repeated block rows.

Control Point	Location in Program
AFTER-ENTER-CHECK	After the end user presses the processing key (the Enter key is the default), and before the PRE-FUNCTION-TEST paragraph executes.
PRE-FUNCTION-TEST	Before APS evaluates all functions except the Terminate, or Exit, function.
PRE-BRANCH	Before each MSG-SW, XCTL, or Call function executes.
ED-ERROR-PRE-SEND	Before APS send a screen whose field edits have failed.
GENERAL-PRE-SEND	After APS checks all functions, and before the TP-SEND call executes, when invocation mode is screen-invoked.
BEFORE-SEND-PARA	Before APS sends the screen, regardless of invocation mode.
PRE-SCREEN-TO-REC	Before APS performs the MOVE-SCREEN-TO-REC paragraph.
POST-SCREEN-TO-REC	After APS performs the MOVE-SCREEN-TO-REC paragraph, and the Update or Add function executes.
PRE-REC-TO-SCREEN	Before APS performs the MOVE-REC-TO-SCREEN paragraph.
POST-REC-TO-SCREEN	After APS performs the MOVE-REC-TO-SCREEN paragraph, and after the Query function executes.
PRE-RB1-ROW-TO-REC	Before the Add or Update function executes for a repeated record block row, and before screen fields move to database fields. APS uses the subscript CTR to reference repeated block rows.
POST-RB1-ROW-TO-REC	Before the Add or Update function executes for a repeated record block row, and after screen fields move to database fields. APS uses the subscript CTR to reference repeated block rows.

Control Point	Location in Program
PRE-REC-TO-RB1-ROW	After the Query or Forward function executes for a repeated record block row, and before database fields move to screen fields. APS uses the subscript CTR to reference repeated block rows.
POST-REC-TO-RB1-ROW	After the Query or Forward function executes for a repeated record block row, and after database fields move to screen fields. APS uses the subscript CTR to reference repeated block rows.
ERROR-SEND-AND-QUIT	When a program terminates abnormally, such as when a database call fails when the Database Call Tailoring screen's Abort On Error parameter is set to y.
LISTBOX-SELECT	When the end user selects a list box item. Enter predefined functions such as *QUERY, *UPDATE, *ADD, and *DELETE.
MISC-USER-PARAGRAPHS	A location where you can write and store multiple paragraphs that you can perform at any control point in your program.

The following flowcharts illustrate the location of the control points in APS-generated programs.

Chart 1: Standard Control Points

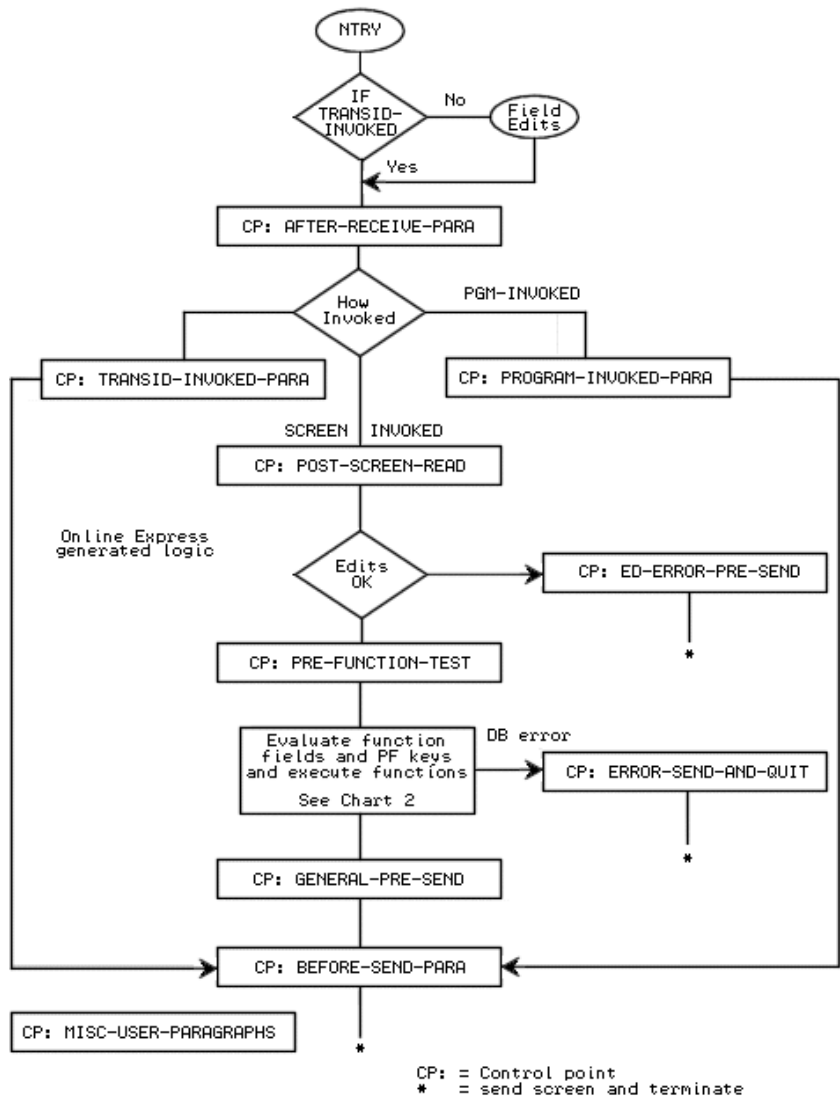
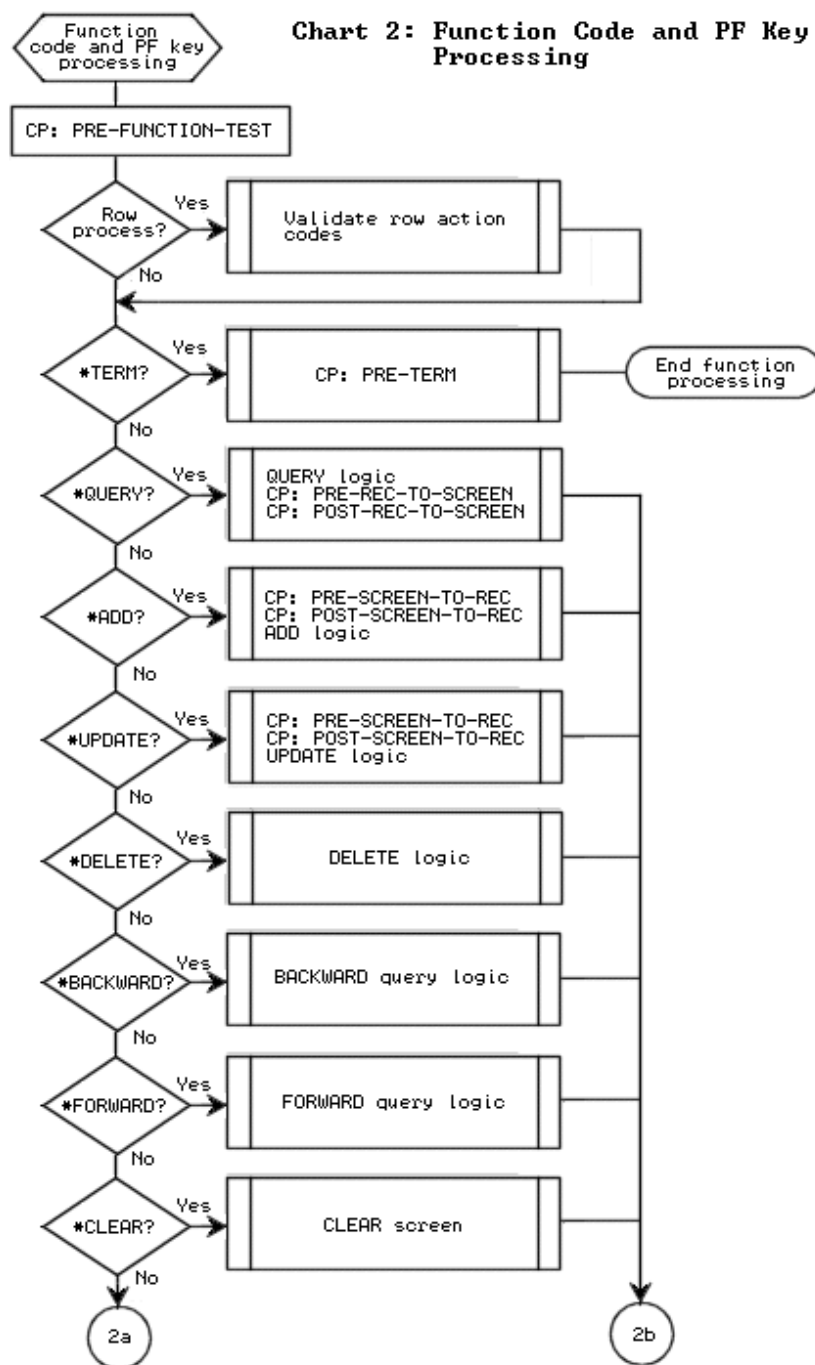
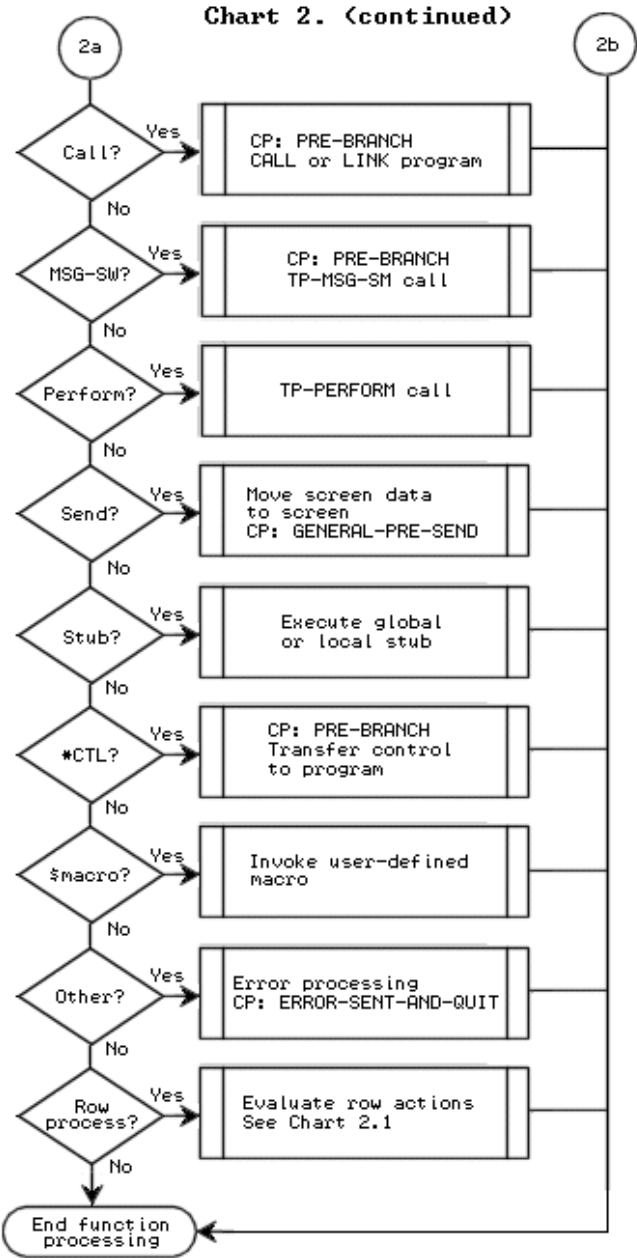
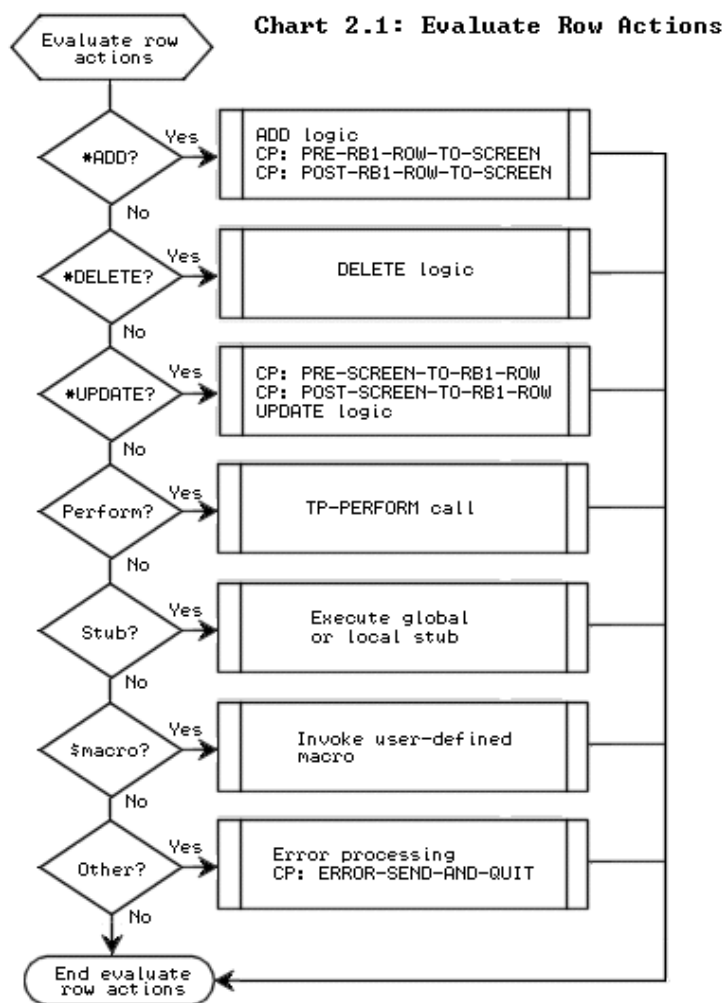


Chart 2: Function Code and PF Key Processing





You can also execute custom logic at any of several APS-provided database call control points, selected from the Database Call Tailoring screen. The database control points are as follows:

Control Point	Location
BEFORE DB ACCESS	Before a non-loop database call executes
BEFORE LOOP	Before a loop database call executes

Control Point	Location
NORMAL STATUS (BEFORE REC MAPPED TO SCREEN)	Before Online Express maps looped records to the screen
NORMAL STATUS	After Online Express maps any records to the screen
EXCEPTION STATUS	After the database call returns a status flag with the Exception status code
ERROR STATUS	After the database call returns a status flag with the Error status code
AFTER DB ACCESS	After a non-loop database call executes
AFTER LOOP	After a loop database call executes

The following flowchart illustrates the location of these control points in APS-generated programs.

Non-Loop Call

CP: BEFORE DB ACCESS

IF OK-TO-PROCEED
 database call
 IF OK-ON-REC

CP: NORMAL STATUS

ELSE-IF exception-status

CP: EXCEPTION STATUS

ELSE

CP: ERROR STATUS

MOVE MESSAGE TO SCREEN
 PERFORM ERROR SEND AND QUIT

CP: AFTER DB ACCESS

Loop Call

CP: BEFORE LOOP

IF OK-TO-PROCEED
 DB-PROCESS call

CP: NORMAL STATUS
 (BEFORE REC MAPPED TO SCREEN)

IF OK-TO-PROCEED
 ADD 1 to CTR
 CHECK CTR
 PERFORM STOREKEY

CP: NORMAL STATUS

PERFORM MOVE REC TO SCRBLKS

CP: AFTER LOOP

15 Reports - APS-Generated

This chapter contains information on APS reports.

Report Descriptions

Application Definition

Display and describe application components painted in the Application Painter. Each application produces a separate report.

Produce from: Painter Menu, Application Painter, or Report Generator screens.

Selection criteria: Current application from the Application Painter screen; specific application, all applications, or range of applications from the Report Generator screen.

Component List

Alphabetically list and total components for each painter.

Produce from: Documentation Facility.

Selection criteria: All painters, or specific painter.

Content Report

Summarize information for each component. Output can be sorted by component name or date.

Produce from: Documentation Facility.

Selection criteria: All painters, or specific painter, specific date or date range. Sort by painter Type, Update date, or Create date.

Data Structure Definition Report

Display and describe data structures from the Data Structure Painter. Each data structure is a separate report.

Produce from: Painter Menu, Application Painter, or Report Generator screens.

Selection criteria: Specific data structure or all data structures in current application from the Application Painter screen; specific data structure, all data structures, or range of data structures from the Report Generator screen.

DDIFILE Report

View contents of DDIFILE.

Produce from: Documentation Facility.

Selection criteria: None.

Entity Cross Reference

Cross reference components and their painters such as other application components that reference the component being cross-referenced.

Produce from: Documentation Facility.

Selection criteria: All, or specific component within a painter. For batch programs, type *BATCH in the Entity Name field and PG in the Entity Type field.

Entity Parts List

List selected parts (components) of a target component (or components).

Produce from: Documentation Facility.

Selection criteria: Component type, *ALL* or specific components, specific application for Entity Type PG; limit of how deep the list goes; APS or user-supplied source components, and/or informational references (default is *ALL*).

Entity Search Utility

List all occurrences in a component (or components) of a specified search expression.

Produce from: Documentation Facility.

Selection criteria for components: Component type, *ALL* or specific components, specific application for Entity Type PG.

Selection criteria for search expressions: Text string of characters or regular expression of text characters and metacharacters, as follows:

Metachar	What it matches
.	Wildcard; matches any character.
()	Denotes a character class containing a range or list of characters. The only metacharacters recognized in a class are ^ as the first character in the class, - to denote a range, and \ to quote a metacharacter.
*	Matches zero or more occurrences of the single character or character class immediately preceding the *.
+	Matches one or more occurrences of either the single character or character class, or any or all characters in the character class immediately preceding the +.
?	Matches zero or one occurrence of the single character or character class immediately preceding the ?.

Metachar	What it matches
^	Code as first character in a character class to indicate logical NOT. Code as first character in expression to match the expression if the match begins at the left margin of the file searched.
\$	Matches the preceding character(s) or character class if match appears at end of line. Must be coded last in expression.
\	Indicates that the single metacharacter immediately following the \ is an ordinary text character, not a metacharacter.

Entity Use Report

List components that use the target component such as components that COPY or INCLUDE the target component.

Produce from: Documentation Facility

Selection criteria: Component type, *ALL* or specific components, specific application for Entity Type *PG*; limit of how deep the list goes; APS or user-supplied source components, or informational references (default is *ALL*).

Field Attributes Report

Summarize field attributes applied to each screen field.

Produce from: Report Generator Options screen.

Selection criteria: Determined by selection criteria of Screen Hardcopy Report.

Field Edit Report

Summarize field edits applied to each screen field.

Produce from: Report Generator Options screen.

Selection criteria: Determined by selection criteria of Screen Hardcopy Report.

Field/Screen Cross Reference

Cross reference screen I/O and text fields and the screens in which they occur.

Produce from: Documentation Facility.

Selection criteria: All fields (press Enter key), specific field, or range of fields.

Macro/Program Cross Reference

List the macros used in one or more applications, along with all of the programs that invoke the macros, and the macro libraries that the programs reside in.

Produce from: Documentation Facility.

Selection criteria: All, or specific application.

Program DB/DC Report

Document screens and PSB/subschemas used by a program. Report has three sections: database views, record I/O areas, and screen I/O areas.

Produce from: Documentation Facility.

Selection criteria: Specific program and subschema, and, optionally, specific screen name(s).

Program Definition Report

Display and describe components of the Program Painter.

Produce from: Painter Menu, Application Painter, or Report Generator screens.

Selection criteria: Specific program or all programs in current application from the Application Painter screen; specific program, all programs, or range of programs from the Report Generator screen.

Report Mock-Up Report

Display report mock-up as painted in the Report Mock-Up Painter.

Produce from: Painter Menu, Application Painter, or Report Generator screens.

Selection criteria: Specific report mock-up or all mock-ups in current application from the Application Painter screen; specific mock-up, all mock-ups, or range of mock-ups from the Report Generator screen. Specify the printing of line numbers on the Report Generator Options screen.

Scenario Definition Report

Display and describe the application scenario as painted in the Scenario Prototype Painter, followed by a display of each screen mock-up in the scenario. Each scenario produces a separate report.

Produce from: Painter Menu, Application Painter, or Report Generator screens.

Selection criteria: Specific scenario, all scenarios, or range of scenarios.

Screen Hardcopy Report

Display screen mock-ups as painted in the Screen Painter. Optionally select reports of attribute values and field edits for each I/O and text field.

Produce from: Painter Menu, Application Painter, or Report Generator screens.

Selection criteria: Specific screen or all screens in the current application from the Application Painter screen; specific screen, all screens, or range of screens from the Report Generator screen. Select Field Attributes Report, Screen Field Edits Report, and left justification for all three reports on the Report Generator Options screen.

Report Production

Application documentation reports may be produced in four ways; some reports can be produced in more than one way. Produce your report by the Produce from method noted previously -- the following describes the procedure for each method.

Report Generator

- 1 On the Painter Menu, enter in the Type field the component type that you want to report on--AP(plication), CN (scenario), DS (data structure), PG (program), RP (report mock-up), or SC(reen).
- 2 Leave the Member field blank.
- 3 Enter REPORT in the Command field and press Enter.
- 4 Enter selection criteria:
 - To report on all members of all component types, type 1 in the Option field. Make sure that all entry fields are blank on the screen.

- To report on all members of the one component type, type 2 in the Option field. Leave the value that displays in the Library field.
 - To report on a specific member of a component, select type 3 in the Option field. Leave the value that displays in the Library field, and enter the component member name in the Member Name field.
 - To report on a range of members in a component, type 3 in the Option field. Enter the component type in the Library field and the value range of the members you want to report on in the Range Greater and Range Less fields.
- 5 Press **Enter**.

Painter Menu

- 1 In the Type field, enter the component type that you want to report on--AP(plication), CN (scenario), DS (data structure), PG (program), RP (report mock-up), or SC(reen).
- 2 In the Member field, enter the member name to report on.
- 3 Enter REPORT in the Command field.
- 4 Press **Enter**.

Application Painter

- 1 To report on all members of all components, or all members of a specific component, do one of the following:
 - Type *REPORT* in the Command field to report on all members of all components of an application.
 - Enter *REPORT* componenttype *ALL* in the Command field to report on all members of a specific component.
- 2 To report on a specific member of a component, do one of the following:
- 3 Enter *REPORT* componenttype componentname in the Command field.

- 4 Type *R* next to the component name in the Application Painter matrix.
- 5 Press Enter.

Documentation Facility

- 1 From the APS Main Menu screen, enter 2 in the Option field.
 - 2 Enter option 2 in the Option field.
 - 3 Enter the applicable option number in the Option field.
 - 4 Enter any selection criteria.
 - 5 Press Enter.
-

16 Reports - Report Writer

This chapter describes Report Writer structures, requirements, and presents mock-ups.

Sample Structure

```
-KYWD- 12-*-----20---*-----30---*-----40---*-----50
IO      Input/Output statements
.
FD      Input FD clause
.
01      Input record description
FD      Output FD clause
.
01      Output record description
RED     reportfilename
        CODE clause
        CONTROL clause
        WRITE ROUTINE clause
        PAGE LIMIT nn LINE
            FIRST DETAIL linenumber
            LAST DETAIL linenumber
            FOOTING linenumber.
MOCK mockupreportname
01      TYPE IS REPORT HEADING /*FOR REPORT HEADER
        MOCKUP LINES clause
        OVERPRINT clause
        SOURCE clause or VALUE clause
01      TYPE PAGE HEADING      /*FOR PAGE HEADER
        MOCKUP LINES clause
        SOURCE clause or VALUE clause
01      TYPE CONTROL HEADING  /*FOR CONTROL HEADER
        MOCKUP LINES clause
        SOURCE clause or VALUE clause
01      TYPE DETAIL           /*FOR DETAIL LINES
        MOCKUP LINES clause
        SOURCE clause or VALUE clause
        REFERENCE clause
```

```

01  TYPE CONTROL FOOTING    /*FOR CONTROL BREAK
    MOCKUP clause
    SOURCE clause or VALUE clause
    SUM clause
NTRY
    .
    INITIATE statement
    .
    GENERATE statement
    .
    TERMINATE statement

```

Requirements

Input-Output Section

Use the IO keyword to code a FILE-CONTROL paragraph with SELECT and ASSIGN clauses.

File-Section

Use the FD keyword and statement to identify input and up to 15 report files.

Report Section

Code the following in the order specified for each report specified in the FD statement:

- Report description entry--Use the RED keyword and statement to specify the report name, control fields, and page characteristics.
- Mock-up identification--Use the MOCK keyword and statement to identify the report mock-up from the Report Mock-Up Painter.

- Group Description Entries--Use the 01 keyword to define the format and characteristics of each report group (headings, detail lines, and footings) using:
 - TYPE statement and clauses to identify the function of the report group.
 - Item description entries (SOURCE, SUM, VALUE, REFERENCE statements) to describe the elements in the group.

Procedure Division

Use the NTRY (or ENTR) or PROC keyword to code Procedure Division statements that:

- Create logic for selecting records.
- Code one INITIATE statement to begin processing all reports named in the FD.
- Code a GENERATE statement for each report named in the FD statement and specify if it is a detail or summary report.
- Code one TERMINATE statement to end report processing for all reports.

Mock-Ups

Use the Report Mock-Up Painter to paint batch report mock-ups. The Report Mock-Up Painter provides full ISPF editing capabilities.

- Define a mock-up by typing the literals and output fields in a visual representation, using columns 1 through 247. In the generated source:
 - Literals become VALUE clauses.
 - Output fields become PIC strings.
- A maximum of 200 lines are allowed per report.
- Use ISPF commands to edit your mock-up.

- Use COBOL edit masks:
- Alphanumeric masks
- Floating numeric formats, including *, \$, +, -, S, 9, Z
- Use PIC clauses for formatted date and time fields

Structures

Keyword

```
-KYWD- 12-*----20---*----30---*----40---*----50---
01      [dataname] TYPE [IS] reportgroup
        Report Writer statements
```

CODE

```
CODE literal
```

CONTROL

```
CONTROL [IS] [FINAL] dataname
CONTROLS [ARE] [FINAL] dataname1 ... datanameN
```

FD Keyword

```
-KYWD- 12-*----20---*----30---*----40---*----50-
FD      reportfilename
        [filedescriptionstatement]
        REPORT IS|REPORTS ARE reptname1 [... reptname15]
```

GENERATE

```
GENERATE dataname|reportname
```

INITIATE

```
INITIATE reportname1 [, reportname2] ...
```

IO Keyword

Format 1:

```
-KYWD- 12-*----20---*----30---*----40---*----50---*----60
IO      COBOLselectstatement
        IOstatement
```

Format 2:

```
-KYWD- 12-*-----20---*-----30---*-----40---*-----50---*-----60
IO      filename ASSIGN [TO] COBOLdataname
        "literal"
        EXTERNAL externalfile
        ORGANIZATION IS RECORD SEQUENTIAL
```

Iterative Expression

Format 1, numerically suffixed data items:

```
item-#number[/limit[/increment]]
```

Format 2, 1-dimensional array:

```
arrayitem (#number[/limit1[/increment1]])
                                                    One subscript range
```

Format 3, 2-dimensional array:

```
arrayitem (#number[/limit1[/increment1]],
                                                    First subscript range
            #number2[/limit2[/increment2]])
                                                    Second subscript range
```

Format 4, 3-dimensional array:

```
arrayitem (#number[/limit1[/increment1]],
                                                    First subscript range
            #number2[/limit2[/increment2]],
                                                    Second subscript range
```

```
#number3[/limit3[/increment3]])  
Third subscript range
```

MOCK Keyword

```
-KYWD- 12-*-----20---*-----30---*-----40---*-----50-  
MOCK   mockupname
```

MOCKUP LINES

```
MOCKUP|M LINE|LINES linenumber1 [THRU linenumberN]
```

OVERPRINT

```
OVERPRINT|O WHEN 'characterstring' AT COLUMN integer
```

PAGE LIMIT

```
PAGE LIMIT IS|ARE number [LINE|LINES]  
[FIRST DETAIL linenumber]  
[LAST DETAIL linenumber]  
[FOOTING linenumber] [.]
```

RED Keyword

```
-KYWD- 12-*-----20---*-----30---*-----40---*-----50-  
RED    reportname
```

REFERENCE

```
R[EFERENCE] [IS] dataname PIC[TURE] [IS] picclause  
[DATA-NAME [IS] sumcounter]
```

SOURCE

```
SOURCE [IS] dataname [iterativeexpression] [PIC picclause]
      [BLANK [WHEN] ZERO]
      [CHANGE INDICATE|GROUP INDICATE]
      [JUSTIFIED|JUST [RIGHT]]
      [DATA-NAME [IS] sumcounter]
```

SUM

```
SUM|+ [IS] dataname1[ iterativeexpression|dataname1] ...
      [UPON detailgroup[ detailgroup] ...]
      [RESET [ON] [FINAL] sumcounter]
      [DATA-NAME [IS] sumcounter]
      [PICTURE|PIC [IS] picclause]
```

SUPPRESS

```
SUPPRESS PRINTING
```

TERMINATE

```
TERMINATE reportname1 [, reportname2] ...
```

TYPE

Format 1, page header:

```
TYPE [IS] PAGE HEADING|PH

      [LINE [NUMBER IS] number      ] [.]
      PLUS number
```

Format 2, page footer:

```
TYPE [IS] PAGE FOOTING|PF

      [LINE [NUMBER IS] number]
```

```
[NEXT GROUP [IS] number      ]  [.]
      PLUS number
```

Format 3, report header:

```
TYPE [IS] REPORT HEADING|RH

      [LINE [NUMBER IS] number      ]
            PLUS number

            number
[NEXT GROUP [IS] PLUS number]  [.]
      NEXT PAGE
```

Format 4, report footer:

```
TYPE [IS] REPORT FOOTING|RF
            number]

      [LINE [NUMBER IS] PLUS number]  [.]
            NEXT PAGE
```

Format 5, control headers and footers:

```
TYPE [IS] CONTROL HEADING|CH [FINAL] controldataname
      CONTROL FOOTING|CF

            number
[LINE [NUMBER IS] PLUS number]
      NEXT PAGE

            number
[NEXT GROUP [IS] PLUS number]  [.]
      NEXT PAGE
```

Format 6, detail lines:

```
detaildataname TYPE [IS] DE[TAIL]

            number
[LINE [NUMBER IS] PLUS number]
      NEXT PAGE

            number
[NEXT GROUP [IS] PLUS number]  [.]
      NEXT PAGE
```

USE

USE BEFORE REPORTING identifier

VALUE

VALUE "characterstring" [PIC picclause]
[DATA-NAME [IS] sumcounter]

WRITE ROUTINE

WRITE ROUTINE [IS] paragraphname

17 Reserved Words

The following words are reserved for APS use:

%	&	--	@
\$	+	++	\$
<+	</	=	<
<<	<*	>	>>
>*	/ *	01	ACCEPT
ACCESS	ADD	ALL	ALTERNATE
APPLY	\$APS-	APSMACS	APSSRC
ARE	ASSIGN	ATTR	AUXOUT
BASIS	BEFORE	BIND	BLANK
BLOCK	BYTES	CA	CALL
CANCEL	CARDIN	CBL	CF
CH	CHANGE	CHARACTERS	\$CIC-
CICS	CLEAR	CLOSE	COBMES
COBIIMES	CODE	CODE-SET	COLUMN
COMMIT	COMP	&COMPILETIME	COMPUTATIONAL
COMPUTE	CONNECT	CONTAINS	CONTINUE
CONTROL[S]	COPY	COPYLIB	CPERFORM
DATA	DATA-NAME	\$DB-	DB-CLOSE
DB-ERASE	DB-IF	DB-MODIFY	DB-OBTAIN
DB-OPEN	DB-ROLLBACK	DB-STORE	DC
\$DDI-	DDISYMB	DE	DEBUG
DECL	DECLARATIVES	&DEFINE[D]	&DEFVAL
DELETE	DEPENDING	DESTINATION	DETAIL
DISCONNECT	DISPLAY	DIVIDE	DIVISION
\$DLG-	DPAR	DS	EDIT-FLAGS
EJECT	ELSE	ELSE-IF	END
END-OBTAIN	ENTER	ENTRY	*EOF*

ERASE	ERROR	ESCAPE	EVALUATE
EXAMINE	EXEC	EXHIBIT	EXIT
FALSE	FD	FILE	FILE-ID
FILE-LIMITS	FILLER	FIND	FINISH
FIRST	FOOTING	FRFM	GENERATE
GET	GO	GOBACK	GROUP
HEADING	HIGH-VALUE	HIGH-VALUES	IDM-
IF	\$IM-	\$IMS-	IN
&INDEX	INDEX[ED]	INDICATE	INITIATE
INPUT	INPUT-OUTPUT	INSERT	INSPECT
IO	IS	JUST	JUSTIFIED
KEEP	LABEL	LAST	LEADING
&LENGTH	LIB1[IN]	LIB2[IN]	LIB3[IN]
LIMIT	LIMITS	LINE[S]	LINE-COUNTER
LINK	LINKAGE	LK	LOW-VALUE[S]
\$MACRO-	MACRO	MAININ	\$MDB-
\$MDC-	MERGE	MOCK	MOCKUP
MODE	MODIFY	MOD-NAME	MOVE
MSG-SW	MULTIPLY	NARROW	NOMINAL
NOT	NOTE	NEXT	NTRY
NUMBER	&NUMERIC	OBTAIN	OBTAIN-BY-KEY
OBTAIN-BY-SEARCH	OBTAIN-NEXT	OBTAIN-PREV	OBTAIN-REL
OBTAIN-REL-BY-KEY	OCCURS	OF	OMITTED
ON	OPEN	OPT	ORGANIZATION
OUTPUT	OVERPRINT	PAGE	PAGE-COUNTER
PARA	&PARSE	PASS	PASSWORD
PERFORM	PF	PFKEY-VALUE	PH
PIC	PICTURE	PLUS	POSTSOUT
PRIVIN	PROC	PROCEDURE	PROCESSING
QUOTE[S]	RDREAD	READ	READY
REC	RECEIVE	RECORD[S]	RECORDING
RD	RED	REDEFINES	REFERENCE
RELEASE	REM	RENAMES	REPEAT
REPORT[S]	REPORTING	RERUN	RESERVE
RESET	RETURN	REWRITE	RF

RH	RIGHT	ROLLBACK	SAME
\$SC-	SC-CLEAR	SCELIB	\$SCP-
\$SCR-	SCRNLIST	SCRSYMB	SD
SEARCH	SECTION	SEEK	SELECT
SEND	SERVICE	SET	SIGN
SKIP1	SKIP2	SKIP3	SORT
SOURCE	SPACE[S]	SPNM	SQL
STANDARD	START	STATUS	STOP
STORE	STRING	STUB	&SUBSTR
SUBTRACT	SUM	SUPRA	SUPPRESS
SYBT	SYDD	SYEN	SYFD
SYLK	SYIO	SYM1	SYM2
SYMBOLIC	SYNC	SYNCHRONIZED	SYRP
SYSDBOUT	SYSIN	SYSOUT	SYSMSG
SYWS	TERM	TERMINATE	TEXT
TIMES	\$TP-	TRACE	TRAILING
TRANCODE-AREA	TRANSFORM	TRUE	TYPE
UNSTRING	UNTIL	UPDATE	UPON
USAGE	USE	USERMACS	USERNAME
VALUE[S]	\$VS-	\$VSAM-	WHEN
WHILE	WITH	WORK1	WORK2
WORK3	WORK4	WORK5	WORK6
WORK7	WORK8	WORK9	WORKING-STORAGE
WRITE	WS	XCTL	ZERO[S][ES]

18 S-COBOL

This chapter describes coding S-COBOL programs, and S-COBOL structures.

Coding Programs

Follow these guidelines when coding programs using S-COBOL

- Indentation controls the logical sequence in which lines of source code are executed.
- Punctuation is unnecessary.
- The first nondeclarative paragraph is executed from top to bottom, and then program execution ends. All other paragraphs are performed from the first paragraph, or from another paragraph that is itself performed.
- The following statements are not valid:
 - *PERFORM ... THRU ... and GO TO ...*
 - *SORT|MERGE THRU ... sectionname with INPUT or OUTPUT PROCEDURE.*
 - *NEXT SENTENCE*
- Code only one verb per source code line.
- Avoid using the double-hyphen (--).
- Enclose S-COBOL literals with either single or double quotes .
- Code Comments as follows:
 - Code anywhere in your program:

```
-KYWD- 12-*----20---*----
/*      commentline
```

- Code in Procedure Division only:

```

-KYWD- 12-*-----20---*-----
        /*commentline

```

- Code in Procedure Division only; do not use with DB/DC calls:

```

-KYWD- 12-*-----20---*-----30---*--
        Program code      /*comment

```

- For macro comments, use the macro comment symbol, %*.
- Code continuation using an ellipsis followed by a space (...).
- Use any of the following relation conditions:

Comparison	Relational Operator
Greater than	IS GREATER THAN IS >
Not greater than	IS NOT GREATER THAN IS NOT > IS <=
Less than	IS LESS THAN IS <
Not less than	IS NOT LESS THAN IS NOT < IS >=
Equal to	IS EQUAL TO IS =
Not equal to	IS NOT EQUAL TO IS NOT =
Greater than or equal to	>=
Less than or equal to	<=

- Use an optional simplified syntax for the following verbs: ADD, COMPUTE, DIVIDE, MOVE, MULTIPLY, SUBTRACT.

Abbreviated Syntax	S-COBOL/COBOL Equivalent
A = B	MOVE B TO A
A B = C	MOVE C TO A B
A = B + C	COMPUTE A = B + C

Structures

AT END/INVALID KEY

```
conditionalverb [AT] END ON filename
                INVALID [KEY] ON filename
statementblock
```

ENTRY

```
ENTRY literal
.
.
[USING identifier1, ..., identifierN]
```

ESCAPE

```
ESCAPE
```

EVALUATE

Format 1:

```
EVALUATE identifier1
WHEN valuexpression1
    statementblock
[ .
.
WHEN valuexpressionN
    statementblockN]

[WHEN OTHER
    statementblock]
```

Format 2 (decision table):

```
EVALUATE identifier1 [, ..., identifierN]
WHEN valuexpression1 [, ..., valuexpressionN]
    statementblock
```

```

[ .
.
WHEN valueexpressionN+1[, ..., valueexpressionN+N]
    statementblock]

[WHEN OTHER
    statementblock]

```

EXIT PROGRAM

```
EXIT PROGRAM
```

IF/ELSE-IF/ELSE

Format 1:

```

IF condition1
    statementblock

[ ELSE-IF|ELSE IF condition2
    statementblock
.
.
ELSE-IF|ELSE IF conditionN
    statementblock]

[ELSE
    statementblock]

```

Format 2:

```

COBOLimperativestatement
... COBOLconditionalclause
    statementblock

ELSE-IF|ELSE IF condition1
    statementblock
[ .
.
ELSE-IF|ELSE IF conditionN
    statementblock]

[ELSE
    statementblock]

```

PERFORM

Format 1:

```
PERFORM paragraphname
```

Format 2:

```
PERFORM paraname (argument1[[,] argument2[,]  
...[,] argumentN])  
.  
.  
paraname ([+|-]argument1[[,] [+|-]argument2[,]  
...[,] [+|-]argumentN])
```

REPEAT

Format 1:

```
REPEAT  
    statementblock  
  
UNTIL|WHILE condition  
    [statementblock]
```

Format 2:

```
REPEAT VARYING|LINKING indexname|identifier1  
... [FROM indexexpression]|arithexpression  
... [BY literal]|identifier2  
  
    [statementblock]  
  
UNTIL|WHILE condition  
    [statementblock]
```

Format 3:

```
REPEAT VARYING|LINKING indexname|identifier1  
... [FROM indexexpression]|arithexpr1  
... [BY literal]|identifier2  
... [DOWN] TO|THRU arithexpr2  
  
    statementblock
```

Format 4:

```

REPEAT VARYING|LINKING clause1
.
.
[... VARYING|LINKING clauseN]

        statementblock1

[UNTIL|WHILE condition
        statementblock]

```

SAGE-TRACE-FLAG

When the SAGE-TRACE-FLAG is tested, APS generates:

```

WORKING-STORAGE SECTION.
.
.
        02  SAGE-TRACE-FLAG PIC X VALUE "T".
.
.
PROCEDURE DIVISION.
.
.
paragraphname
        IF SAGE-TRACE-FLAG = TRUE
            DISPLAY "EXEC:--paragraphname--".
.
.

```

SEARCH

Format 1:

```

SEARCH identifier1 VARYING indexname|identifier2
... [[AT] END]
        statementblock

[WHEN searchcondition1
        statementblock

.
.

```

```
[WHEN searchconditionN
      statementblock]]
```

Format 2:

```
SEARCH ALL identifier [[AT] END]
      statementblock
```

```
WHEN searchcondition
      statementblock
```

STOP RUN

```
STOP RUN
```

SUPRA

S-COBOL source input:

```
WS
01  [viewname] INCLUDE logicalviewname [(userfieldlist)]
    .
    .
    SUPRA suprastatement
    ...  suprastatement continuation
    ...  .
    ...  .
```

COBOL source output:

```
WORKING-STORAGE SECTION.
01  [viewname] INCLUDE logicalviewname [(userfieldlist)]
    .
    .
    suprastatement
    suprastatement continuation
    .
    .
```

TRUE, FALSE

```
TRUE|FALSE dataname1 [dataname2 [... datanameN]]
```

TRUE/FALSE/ ALWAYS/NEVER

APS generates in Working-Storage:

```
01 GENERATED-FLAGS.  
  02 TRUX          PIC X  VALUE 'T'.  
    88 ALWAYS      VALUE 'T'.  
    88 NEVER       VALUE 'F'.  
  02 FALSX         PIC X  VALUE 'F'.
```

UNTIL/WHILE

```
UNTIL|WHILE condition1 [AND|OR condition2  
... [... AND|OR conditionN]]  
  
    statementblock
```

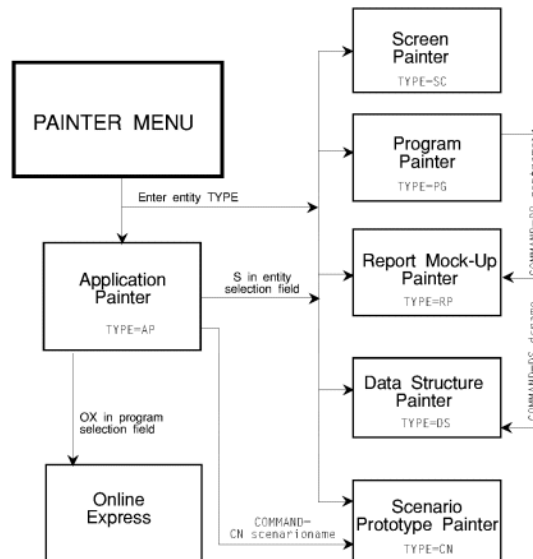
USERNAME

```
USERNAME paragraphname
```

19 Scenario Painter

This chapter contains information on navigating in the Scenario Painter, and defining and running a scenario.

Painter Navigation



Defining and Running a Scenario

Use the Scenario Painter screen to define a sequence of screens for scenario prototyping.

- 1 Access from the Application Painter. All application screens automatically display.
- 2 Use the Retitle command to display screen titles previously defined in Screen Painter.
- 3 Enter User Comments of up to 18 characters to display during prototyping.
- 4 To run the scenario, enter RUN in the Command field.
- 5 Use any of the commands when running the scenario. Enter the command in the upper left corner.

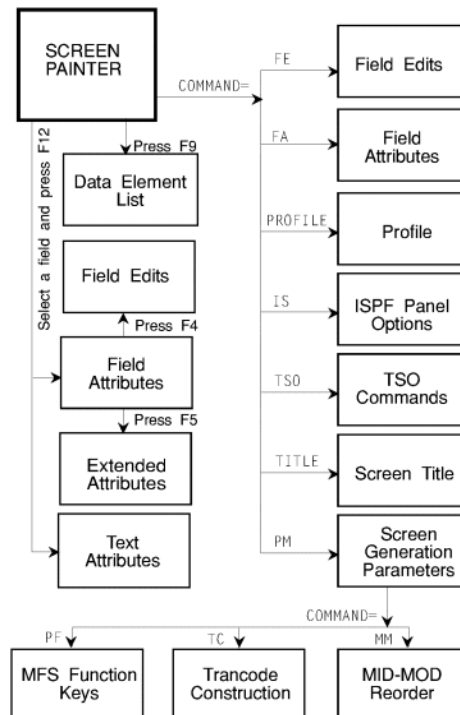
Command	Description
<i>+increment</i>	Display screen increment screens ahead.
<i>-increment</i>	Display screen increment screens behind.
<i>DATA</i>	Prepare scenario screen for data entry and display.
<i>DATAOFF</i>	Turn off simulation and display screen in its painted format.
<i>EDIT</i>	Display Screen Painter to create or modify a screen.
<i>END CANCEL QUIT</i>	Terminate scenario; transfer to screen that invoked the scenario.
Enter key	Display next screen in scenario.
<i>FIRST START</i>	Display first screen in scenario.
<i>HELP F1 F13</i>	Display Help tutorial.
<i>LAST FINAL</i>	Display last screen in scenario.
<i>NUM [ON]</i>	Display descriptive information on last line of scenario screen.
<i>NUM OFF</i>	Turn off NUM ON command. Default when scenario invoked.
<i>READ</i>	Display data saved by last SAVE command.

Command	Description
SAVE	During prototype, save data entered in screen fields.
<i>screenname</i>	Display named screen.
<i>+screenname</i>	Display next occurrence of named screen.
<i>-screenname</i>	Display last occurrence of named screen.
<i>screennumber</i>	Display screen with sequential position of screennumber.

20 Screen Painter

This chapter contains information on screen navigation, screen creation and design options, field attributes, field edits, modifying and generating screens, and IMS and ISPF Dialog support functions.

Screen Navigation



Creating Screens

Use the Screen Painter to create and modify screens, and to assign field attributes, field edit functions, and generation parameters. Some guidelines are:

- Paint initial mock-up as end user will view it with Xs (I/O fields) or literal text (text fields); leave all other positions blank (spaces).
- Column 1 is reserved for attribute bytes in the generated screen definition. It is not available for screen entry.
- If literal text consists of "stand-alone" Xs, the space immediately preceding or following the Xs should contain an underline character.
- When using ampersands (&) under ISPF, leave as many spaces at the right of the line as there are ampersands on that line.

Screen Design Options

To specify editing session options, type profile in the Command field. Select editing session options, as follows:

Option	Description	
Command Location	Specify where the Command field appears--enter <i>TOP</i> for the top-left corner or <i>BOTTOM</i> for the bottom-left corner.	
Caps on/off	<i>ON</i>	Convert text fields to upper case.
	<i>OFF</i>	Preserve or restore text fields as you enter them.
Nulls on/off	<i>ON</i>	Insert data directly into a row.
	<i>OFF</i>	Fill rows with spaces.
Keys on/off	<i>ON</i>	Display the Screen Painter function key definitions at the bottom of the screen.
	<i>OFF</i>	Do not display keys.

Option	Description	
Display field name	YES	Activate the Field Name screen. As you move the cursor between fields by pressing the Enter key, this screen displays the name of the current field.
	NO	Do not display the field name.

Field Attributes

To assign field attributes, position cursor on specific I/O field or text field and press PF12, or enter FA in command field to transfer to Field Attributes screen.

Attribute	Description and Values	
Name	I/O field name; maximum 16 characters.	
Length	Display field only; to change field length, move the cursor to the Xs designating the field and space over or delete the Xs representing the field, or extend the field with more Xs.	
Intensity	B	Bright
	N	Normal (default)
	D	Dark.
Type	U	Unprotected (default); field is for both input and output.
	P	Protected; field is output only.
MDT	Applies to IMS and CICS only.	
	ON Default. Always keep modified data tag on, whether or not the end user modifies field; default for I/O fields.	
	Turn tag on when end user modifies field; default for text fields.	
Value	Initial value for screen field; maximum is field length or 27 characters, whichever is less.	
Num Lock	ON	Activate keyboard numeric shift lock
	OFF	Deactivate numeric shift lock (default)

Attribute	Description and Values	
Light Pen	ON	Light pen detectable.
	OFF	Not light pen detectable (default).
Init cursor	NO	Do not position cursor on this field when the program sends the screen. Default for all but the first I/O field.
	YES	Position cursor on this field. Default for first I/O field.
Color	B	Blue
	G	Green
	Y	Yellow
	N	Neutral
	P	Pink
	R	Red
Highlight	T	Turquoise
	B	Blinking
	U	Underline
Modify	R	Reverse video
	IMS only.	
	NO	Program cannot modify extended attributes at run time (default).
	YES	Program can modify extended attributes. APS generates the extra attribute bytes required.

Field Edits

To assign field edits do the following:

- 1 In the Screen Painter, enter FE in the Command field. Or, from the Field Attributes pop-up screen, press PF4, or enter FE in the Command field.
- 2 From the Field Selection screen, transfer to an edit specification screen by entering one of the options displayed on the screen to the

left of the field name. Or, from the Edit Selection screen, enter an S next to the applicable category. Select an option as follows:

Option	Description
Edit Selection	Display the Edit Selection screen to see a summary of edits for that field and transfer to other edit specification screens.
Internal Picture	Display the Internal Picture screen to specify the internal storage format.
Input Editing	Display the Character Input or Numeric Input screen, depending on whether the internal picture specification is character or numeric, to assign input field edits.
Error Processing	Display the Error Processing screen to specify error messages and attributes to display when the data for the field fails input edits.
Application Edits	Display the Application Editing screen to create your own edit routine to process input data.
Output Editing	Display the Character Input or Numeric Input screen, depending on whether the internal picture specification is character or numeric, to assign output field edits.
Values Or Conversions	Display the Values or Conversion screen to specify a valid value or range of values for input data, or conversion values for either input or output data.
Special Edits	Display the Special Edits screen to select predefined date, user-defined date, of time field edit specifications.

3 Enter values as appropriate.

Application Edit Routines

Select the Application Editing prompt on any Field Edit screen and complete the following options:

Option	Description
Type	Indicate whether this application edit is a paragraph, subprogram, or APS macro. Default is P(aragraph).
Name	Enter a descriptive name for the application edit; maximum 32 characters.
Arguments	To pass a screen field or error flag, prefix the name with the screen name and a hyphen. Enter the following arguments separated by commas and enclose literals in single quotation marks:
Paragraph	Data names or literals that pass to the paragraph through a PERFORM with arguments statement in the generated program.
Subprogram	Data names that appear on the CALL USING statement in the generated program.
APS macros	Customization Facility macro terms, literals, and numeric literals. Do not enclose arguments with double quotation marks.
Execute Before/After APS Edits	Specify when the program executes this application edit: before or after the normal APS field edit routine. Default is B(efore).
Paragraph COPYLIB or APS Macro USERMACS Member	Specify an associated COPYLIB member or a paragraph or the associated USERMACS member name for an APS macro.
Working-Storage COPYLIB Member	Specify a Working-Storage COPYLIB member to be included in the program Working-Storage section.

Alternately, select a predefined edit from the Application Edit List as follows:

- 1 On the Application Editing screen, enter APPLLIST listname in the Command field, where listname is the name of the list of available edits.
- 2 Enter S before the input or output edit routine you want.
- 3 Press Enter to select the edit. You can then modify the selected edit routine.

Bypassing Field Edits

- 1 Access the Field Selection screen, and enter PM in the Command field.
- 2 On the Parm screen, press Enter.
- 3 Define bypass conditions as follows:

Field	Description and Values
Field Name	Specify any field on the screen, including a field in a repeated block, to bypass.
Value(s)	Specify the value or values that let end users bypass input edits. Valid COBOL reserved words are SPACES, LOW-VALUES, and HIGH-VALUES.
Additional Value(s)	Enter as many additional bypass values that can fit on the line; separate each value with a comma.
Program Function Keys	Type S in the selection field to indicate which PF keys the end user can press to bypass the input edits.

Character Field Edits

Internal Option	Description and Values
Data Type	A Alphabetic field. C Default. Character field. G For KANJI use only. Extended Graphics Character Set (EGCS).
Internal Length	Enter the number of characters. The default is the screen field length.
Justified Right	Enter S to generate right justification on the COBOL picture.
Input Option	Description and Values
Internal Picture	Enter S to transfer to the Internal Picture screen to change the storage format.
Required	Enter S to indicate that the end user must enter a value in the field.
Input Mask	Enter the pattern or mask to accept input data and separators. Or, enter S to transfer to the Masking screen and specify the mask in the Input Mask field.
Minimum Input	Enter the length of the shortest valid entry. Default is zero.
Maximum Input	Enter the length of the longest valid entry. Default is field length.
No Embedded Spaces	Enter S to reject characters separated by spaces.
Numeric Test	Enter S to allow only numeric data.
Output Option	Description and Values
Internal Picture	Enter S to transfer to the Internal Picture screen to change the storage format.
Output Mask	Enter the pattern or mask to position data and separators. Or, enter S to transfer to the Masking screen and specify the mask in the Output Mask field.

Output Option	Description and Values
Output Picture	Enter the output COBOL picture. Or, enter S to transfer to the Output Picture screen and specify the mask in the Picture field. You cannot assign an output picture if you use an output mask.
Right Justify	Enter S to generate right justification for the output format.

Copying Field Edits

- 1 Access the Edit Selection screen for the field you are copying field edits to.
- 2 Enter COPY in the Command field.
- 3 Enter the field name you are copying edits from; it must be the same length as the current field. The current field inherits the edits of the copied field, and loses any prior edits.

Conversion Values

- 1 Select the Values or Conversions prompt on any Field Edit screen.
- 2 Assign conversion values using the following syntax format:

`(input1, input2, ..., inputN, I=internalvalue,
O=outputvalue)`
- 3 To specify that the listed conversions are the only valid input values, select the Verify Conversion Values option.

Date Fields--Predefined

Field	Description and Values
Storage Format	J Julian JP Julian packed G Gregorian

Field	Description and Values
	GP Gregorian packed If you do not specify a Storage Format, select the System Date Display option.
In/Out Format	Enter the format number. Formats 1 through 6 are valid input formats. Formats 1 through 4 contain two styles; the field length determines which is assigned.
Date Required	Enter S to indicate that the end user must enter a value in the field.
Error Processing	Enter S to transfer to the Error Processing screen.
System Date Displayed	Enter S to capture the system date.

Date Fields--User-Defined

Field	Description and Values
Internal Picture	Enter s to transfer to the Internal Picture screen to specify the COBOL picture characteristics.
Storage Format	Specify the internal storage format mask. This format must equal the Internal Picture length.
Input Format and Output Format	Valid mask characters are Y (year), M (month), D (day), and special characters if the field is not defined numeric. Restrictions are: Y Can be 2 or 4 characters M Can be 2, 3, or 9 characters D Can be 3 characters if you do not define Month, otherwise it must be 2 characters Month, day, and year must be in the same order in both the Input and Output Formats.
Date Required	Enter s to indicate that the end user must enter a value in the field.

Field	Description and Values
System Date Data	<p>Enter one of the following:</p> <ul style="list-style-type: none">I During input editing, insert the system date only if the field is blank.IR During input editing, always insert the system date, regardless of the field contents.O During output editing, insert the system date if the field is blank.OR During output editing, always insert the system date, regardless of the field contents.

Deleting Field Edits

- To delete all field edits for all fields on the screen, access the Field Selection screen, and then enter DELETE ALL in the Command field.
- To delete all field edits for a specific field, access the Edit Selection screen for the field, and enter D after the Internal Picture prompt.
- To delete a specific field edit for a specific field, access the Edit Selection screen for the field, and enter D to the right of the field edit name.

Error Messages - Field-Specific

- 1 Select the Error Processing prompt on any Field Edit screen.
- 2 Code messages of up to 75 characters. This message overrides any global default messages you created.
- 3 Alternately, copy the default messages and attributes from the Parm screen by enter DEF in the Command field. Modify the message as desired.

Error Messages--Global for All Fields

- 1 Access the Field Selection screen, and enter PM in the Command field.
- 2 On the Parm screen, type the name of the field to display the error message.
- 3 Enter the text to display for invalid data.
- 4 Enter the text to display for not data when data is required.
- 5 Specify the attribute values for fields in error; the default assigns bright and cursor positioning on the field.

Numeric Field Edits

Internal Option	Description and Values
Data Type	Specify N for numeric field.
Internal Length	Enter the number of digits that precede the decimal point. The default is the screen field length.
Decimal Places	Enter the number of digits that follow the decimal point.
COMP (Binary)	Type S to store input data in binary format. Not valid with signed data.
COMP-3 (Packed)	Type S to store input data in packed format. Not valid with signed data.
Signed	Type S to store the input data with either a positive or negative value. Not valid with a binary or packed format.
Sign Leading	Type S to store the sign at the left of the number. Not valid with a binary or packed format.
Sign Separate	Type S to store the sign in a separate byte from the number. Not valid with a binary or packed format.

Input Option	Description and Values
Internal Picture	Enter S to transfer to the Internal Picture screen to change the storage format.
Input Mask	Enter the pattern or mask to accept input data and separators. Or, enter S to transfer to the Masking screen and specify the mask in the Input Mask field.
Numeric De-Edit	Enter S to validate that the data is numeric and to remove special characters. Select this option if you specify an output COBOL picture.
Required	Enter S to indicate the end user must enter a value in the field.
Zero When Blank	Move zero to the internal picture if no data is entered. Not valid with required fields or fields with input masks.
Minimum Digits	Enter the smallest number of digits required before the decimal point. Default is zero. Not valid with an input mask.

Field	Description and Values
Minimum Decimals	Enter the smallest number of digits required to follow the decimal point. Default is zero.
Maximum Digits	Enter the largest number of digits allowed before the decimal point. Default is the maximum number that fits in the internal picture. Not valid with an input mask.
Maximum Decimals	Enter the largest number of digits allowed to follow the decimal point. Default is the maximum number that fits in the internal picture.

Output Option	Description and Values
Internal Picture	Enter S to transfer to the Internal Picture screen to change the storage format.
Output Mask	Enter the pattern or mask to position data and separators. Or, enter S to transfer to the Masking screen and specify the mask in the Output Mask field.

Output Option	Description and Values
Output Picture	Enter the output COBOL picture. Or, enter S to transfer to the Output Picture screen and specify the mask in the Picture field. You cannot assign an output picture if you use an output mask.
Insert Comma(s)	Enter S to format data with commas in appropriate positions.
Zero Suppression	Enter S to generate the zero suppression symbol.
Floating Symbol	Enter a \$, +, or - symbol to generate a floating dollar, plus sign, or minus sign to the left of the first digit.
Fixed Leading Symbol	Enter a \$, +, or - symbol to generate a fixed dollar, plus sign, or minus sign in the leftmost position.
Fixed Trailing Symbol	Enter a \$, +, or - symbol to generate a fixed dollar, plus sign, or minus sign in the rightmost position.

Range of Values

- 1 Select the Values or Conversions prompt on any Field Edit screen.
- 2 Assign a specific value or a range of values using the following syntax formats:

```
value
lowvalue TO|THRU highvalue
lowvalue UP
highvalue DOWN
```

Time Fields

Field	Description
Storage Format	Specify the internal storage format mask. Valid characters are HH (hour), MM (minute), SS (second), *s and special characters if the field is not defined numeric. The storage format must equal the Internal Picture length.

Field	Description
Input Format and Output Format	Type the input and output format masks. Valid mask characters are HH (hour), MM (minute), SS (second), *s, and special characters if the field is not defined numeric. Append your mask with asterisks to indicate the AM and PM indicators that you specify, using a one-to-one correspondence. Hours, minutes, and seconds must be in the same order in both the Input and Output Formats.
Input Required	Enter s to indicate that the end user must enter a time value in the field.
System Time Data	Enter one of the following: <div><div>I</div>During input editing, insert the system time only if the field is blank.</div> <div><div>IR</div>During input editing, always insert the system time, regardless of the field contents.</div> <div><div>O</div>During output editing, insert the system time if the field is blank.</div> <div><div>OR</div>During output editing, always insert the system time, regardless of the field contents.</div> <div>*** End of nested table ***</div>
AM Indicator	Type the AM time indicator, such as a.m. or am.
PM Indicator	Type the PM time indicator, such as p.m. or pm.

Modifying Screens

Change Field Length

Delete or space over Xs or extend field with additional Xs.

Create a Block

- 1 Position cursor on top row of row(s) to be duplicated. Press PF7.
 - 2 Complete fields in Repeated Block screen. Press Enter.
-

Modify a Block

- 1 Position cursor within block. Press PF7.
 - 2 Select option from Repeated Block Menu. Press Enter.
-

Move Field(s)

I/O field:

- 1 Place cursor on field. Press PF4.
- 2 Use arrow keys to position cursor at receiving destination Press PF4.

Text field:

- 1 Press PF10 at left boundary of text field. Press PF10 at right boundary of text field.
 - 2 Use arrow keys to position cursor at receiving destination. Press PF4.
 - 3 To move several fields in a row, see *Move Part of a Row*.
-

Move an Entire Row

- 1 Place cursor on row (but not on an I/O field) to be moved. Press PF4.
 - 2 Use arrow keys to position cursor at receiving destination. Press PF4.
-

Move Part of a Row

- 1 Press PF10 at left boundary of row section to be moved. Press PF10 at right boundary of section.

- 2 Use arrow keys to position cursor at receiving destination. Press PF4.

Copy Field(s)

I/O field:

- 1 Place cursor on row (but not on an I/O field) to be moved. Press PF4
- 2 Use arrow keys to position cursor at receiving destination. Press PF5.

Text field:

- 1 Press PF10 at left boundary of text field. Press PF10 at right boundary of text field.
- 2 Use arrow keys to position cursor at receiving destination. Press PF5.

Copy an Entire Row

- 1 Place cursor on row (but not on an I/O field) to be moved.
- 2 Use arrow keys to position cursor at receiving destination. Press PF5.

Copy Part of Row

- 1 Press PF10 at left boundary of row section to be moved. Press PF10 at right boundary of section.
- 2 Use arrow keys to position cursor at receiving destination. Press PF5.

Delete an I/O Field

Space over field to be deleted or position cursor on field to be deleted and press PF6 twice.

Delete Row

Space over field to be deleted or position cursor on row (but not on an I/O field) to be deleted and press F6 twice.

Screen Generation Parameters

- 1 From the APS Screen Painter, enter PM in the Command field.
- 2 Change parameter values as desired. Parameter values of T(rue) and Y(es) are interchangeable, as are F(alse) and N(o).

All Targets

Parameter	Description and Values	
Prt Asm Mac Expn	F	Default. Do not print expanded assembler macros.
	T	Print macros.
No Assembler END	F	Default. Do not generate an assembler END statement.
	T	Generate statement.
Retain Datanames	F	Default. Do not retain painted field names as assembler labels.
	T	Retain field names. Under BMS or MFS, duplicate or invalid names can occur due to the maximum number of characters that BMS and MFS allow.
Exattr Modifble	F	Default. Do not modify extended attributes at run time.
	T	Allow modification at run time; generate EXTATTR=YES and extra attribute bytes in DSECT.

Parameter	Description and Values
SYSMMSG	NO or blank Default. Do not display system messages.
	YES or SYSMMSG Display messages on last line of the screen, if space is available.
	fieldname Display messages in fieldname.
	YES, row, len Display message of up to length characters on specified row. Row default is last line of screen.
	YES, row YES, length Length can be from 40 to 70 characters or up to 131 characters for MOD5 screens.
Intensity	Change the intensity of all text fields.
	N Default. Normal
	B Bright
Color	Change the color of all text fields.
	NU Neutral
	BL Blue
	PK Pink
	RD Red
	GN Green
	YL Yellow
Blink Rvideo Underline	TQ Turquoise
	Set only one field to T(rue) for text fields.
	Blinking, reverse video, and underline are mutually exclusive.

CICS Parameters

Parameter	Description and Values
Associated Trans	Specify an associated transaction ID; default is the first four characters of the screen name.

Parameter	Description and Values
Mapset Name	Override an APS-generated name; maximum seven characters. To generate a multiple-map mapset that includes some or all screens, assign the same mapset name to the applicable screens in the application. The default mapset name reflects the number of characters in the screen name, as follows: 4-char name: screennameSET 5-, 6-char name: screenname\$ 7-char name: screenname\$; the \$ replaces the 7th char
Line	Starting line of the map on the physical screen; default is 001; value cannot exceed the screen depth.

ISPF Prototype Parameters

Parameter	Description and Values
Global Fld Unpro	F Default. Do not unprotect all I/O fields for prototyping. T Unprotect all I/O fields.
Associated Pgm	Name of the program receiving control from the screen; default program name is screenname.

IMS DC Parameters

Parameter	Description and Values
Device Type	Standard device characters for different model terminals and printers. Defaults are IBM-recommended device characters.
Cursor Feedback	F Default. Do not define a field in the MID as the cursor feedback field. T Provide cursor information for input processing.

Parameter	Description and Values
DIF-DOF Name	<p>Override APS-generated name. Default reflects the number of characters in the screen name, as follows:</p> <p>4-char name: screennameDF 5-, 6-char name: screenname\$ 7-, 8-char name: screenname truncated to 6 characters</p>
Parameter	Description and Values
Opr Logical Paging	<p>F Default. Do not request operator logical paging.</p> <p>T Request paging. Enter name of field that will contain the paging requests in the Optional Fld Name field.</p>
MID Name	<p>Override APS-generated name. Default reflects the number of characters in the screen name, as follows:</p> <p>4-char name: screennameMI 5-, 6-, 7-char name: screennameI 8-char name: screennameI; I replaces the 8th char</p>
MID Default Values	<p>F Default. Do not treat initial values as default values for fields in the MFS-generated MID.</p> <p>T Treat initial values as default values.</p>
MOD Name	<p>Override APS-generated name. Default reflects the number of characters in the screen name, as follows:</p> <p>4-char name: screennameMO 5-, 6-, 7-char name: screennameO 8-char name: screennameO; O replaces the 8th char</p>
MOD Fill Char	<p>Generate fill characters in the MOD segment statement. Valid characters are: --, NULL, PT, C, or any character value.</p>
DSCA	<p>Override the Default System Control Area default value of X'00A0'.</p>

Parameter	Description and Values
"Labeled" Screen	F Default. Do not append screen name to the input message. T Append the screen name.
Lines Per Page	If device type is a printer, specify number of lines to print on a page.
Trancode: Literal	Specify any literal value as the tranocode. Default is the screen name.
Optional Fld Name	Specify fieldname or MFS PFKEY to hold the tranocode or operator logical paging command. Alternatively, enter *PF and assign the PF key value on the MFS Function Keys screen, or *TC and construct a tranocode on the Trancode Construction screen.
MID Segment Exit: Number Vector	Generate the EXIT parameter on the MID segment statement with Number as the exit routine number and Vector as the exit vector number. Valid values are: Number 0 to 127 Vector 0 to 255

IMS Support Functions

MID MOD Reordering

Arrange fields in the input message in an order other than their physical order on the screen as follows:

- 1 From the Screen Generation Parameters screen, enter MM in the Command field.
- 2 Enter new sequence numbers in Order field
- 3 Enter REORDER in the Command field.

PF Key Assignment

Assign trancodes, IMS commands, or logical paging commands to PF keys as follows:

- 1 On the Screen Generation Parameters screen, enter **PF* in the Optional Fld Name field.
- 2 Enter PF in the Command field.
- 3 Enter **NULL* or **SPACE* in the PFkey Global Default field.
- 4 Enter PF key functions.

Trancode Construction

Create a tranocode as follows:

- 1 On the Screen Generation Parameters screen, enter **TC* in the Optional Fld Name field.
- 2 Enter TC in the Command field.
- 3 Complete the screen fields.

Field	Description and Values
Field Name	Enter name of field or *PF. If you enter *pf, specify PF key values on the MFS Function Keys screen.
Default Literal	Enter any literal.
Fill	Enter the MID fill character for the associated field name or *PFKEY, as follows: S Default. Space N Null

ISPF Dialog Support Functions

Generate native panel definition statements for the ISPF Dialog environment,as follows:

- 1 From any screen in the Screen Painter, enter IS in the Command field.
- 2 Complete the screen fields.

Field	Description and Values
Command Field	Enter screen field name for the panel command field that allows end users to enter ISPF commands and prevents truncation errors when they use PF keys. Default is the first unprotected I/O field on the screen. This option generates the native)BODY CMD(variable) statement.
Long Message Short Message	Enter SYSMMSG, if specified on Screen Generation Parameters screen, or a screen field name for the panel long and short message fields. These options generate the native)BODY LMSG(variable) statement for the long message and the)BODY SMSG(variable) statement for the short message.
Help Panel	Enter the name of the panel to display, if the end user requests help. This option generates the .HELP = panelname statement.
Pfkey Option	P Program controls PF key processing. When the program terminates, the original PF key values are restored. I Default. ISPF controls PF key processing.

21 User Help

This chapter describes creating user help and revising help text.

Creating

Use any of the following User Help source utilities to create your help source file.

Applications Utility	Create a complete application help system in one session.
Screens Utility	Create help for individual screens, as well as field help, field value selection lists, and messages.
Data Elements Utility	Create help for global fields that reside in the APS Data Element Facility.

Create the help source file as follows:

- 1 From the APS Main Menu, enter option 2 in the Command field. Then, enter option 4 in the Command field.
- 2 Select enter 1 in the Command field.
- 3 Select a utility to create your help source file and complete the fields on the screen.

Field	Screen	Description	
Context	Data Elements	Enter a context name or press Enter to display a selection list.	
Context List	Data Elements	NO	Do not create a context list.
		YES	Create a context list.

Field	Screen	Description	
Application name	Applications	Enter the application name. Press Enter to display a selection list.	
Field name	Data Elements	Enter the field name or press Enter to display a selection list.	
Screen name	Screens	Enter the screen name or press Enter to display a selection list.	
Edit business name	All	NO	Defaults to the application name.
		YES	Assign a descriptive business name.
Edit text	All	NO	Do not create help text.
		YES	Create help text.
Include screens	Applications	NO	Do not create screen help.
		YES	Create screen help.
Include fields	Applications	NO	Do not create field help.
		YES	Create field help.
Local Fields	Applications and Screens	NO	Do not create local field help.
		YES	Create field help
Create Values	All	NO	Do not create field value help.
		YES	Create field value help.
Help source file name	All	Help source file name, C:\TMP\APSEXT.	
<hr/>			
Note: Do not use an extension with this file name.			

- 4 Press Enter.
- 5 If you entered YES in the Edit Business Name field, enter a business name with a maximum of 55 characters and press F3.
- 6 If you entered YES in the Edit text field, enter up to 50 lines of help text, 73 characters per line. Edit text using ISPF line commands. If you create a field value selection list, enter one value per line. If your end users will create the help text, press F3 to leave this screen blank.
- 7 Specify the database target of your application in the APHLPIN control file in the APS CNTL member.
- 8 Compile the APS provided batch program, A1UHUPD, in the APSUHELP application.
- 9 To execute A1UHUPD to update the help database, from the APS Main Menu, enter 2 in the Command field on the APS Main Menu and then enter 6 in the Command field on the Dictionary Services menu.
- 10 Enter 2 in the Command field and complete the fields as follows:
- 11 To store help text that has been changed during a specific time frame, enter values in the Date and Time fields. To store help text for all application components, leave these fields blank.
- 12 In the Input Help Source File Name field, specify the name of the help source file that you want to store in the help database.
- 13 If the help database is VSAM, enter the name of the help database in the User help database name field.
- 14 Enter your DC/DB Targets
- 15 If the help database is VSAM, enter the DDNAME of the help database in the Environment name field and press Enter.

Revising Help Text

Edit the text of a help database record as follows:

- 1 In CICS, type *apsf*. In IMS, type */for apsfmo*. In ISPF, you execute APSFM using the APS execution facilities. From the APS Main Menu, enter 3 in the Command field. APS displays the Prototype Execution screen. From this screen, enter 1 in the Command field. From the Prototype Execution screen, enter 1 in the Command field. Type *apsfm* in the program field. APS displays the Help Database Maintenance screen.
- 2 Enter APSFM in your system Command field. and enter (A)pplication help, D (Field help), (M)essage, (S)creen help, or V (Field value selection list) in the Entity Type field.
- 3 Complete the remaining fields on the Help Database Maintenance screen as follows:

Entity Type	Description
A	Enter the application name in the Name field and press F5.
M or S	Enter the screen name in the Name field and Press F5.
D or V	Enter the screen name in the Entity Global Type field and the field name in the Name field. If the field is a global data element, enter the data element context name in the context name field and press F5.

- 4 Edit the text using the ISPF line command keys listed on the screen.
- 5 Press F10 to save the help database record, or F3 to save the record and exit the program.

Index

Symbols

\$IM-calls 40
++ keyword 113

Numerics

01 keyword 113
66-level number 84
88-level number 84

A

Alternate Functions screen 127
application development
 APS-generated report 169
application development steps 8
Application Painter
 APS-generated report 169
 limits 122
 navigating 17
 screen 17
APS Painters
 navigating 7, 17
AT END statement 195
ATTR call
 CICS 58
 IMS DC 65
 ISPF Dialog 70
 ISPF prototyping 74

B

BACKOUT call
 CICS 64
 IMS DC 69
BEGIN statement 26
BIND call
 IDMS DB 32
browsing APS components 13
built-in functions 26
 SQL 41
bypassing field edits
 screen fields 211

C

CA keyword 114
character field edits
 screen fields 212
Checkin screen 87
Checkout screen 89
CIC- calls 58
CICS
 control file 23
 data communications calls 58
 screen generation parameters 223
CLEAR call
 CICS 61
 IMS DC 65
 ISPF Dialog 70
CLEAR-ATTRS call
 CICS 61
 IMS DC 65

- CLOSE call
 - IDMS DB 32
 - SQL 42
 - VSAM batch 50
 - COBOL/2
 - limits 121
 - Column Selection screen 129
 - COMMIT call
 - SQL 42
 - COMPILETIME function 26
 - control files 23
 - control points 130, 158
 - conversion values
 - screen fields 213
 - Convert command 86, 119
 - copying
 - field edits 213
 - screen fields and rows 221
 - Correlation Names screen 131
 - creating
 - application components 13
 - Customization Facility
 - limits 121
 - structures 25
- ## D
- data communications calls 57
 - CICS 58
 - IMS DC 65
 - ISPF Dialog 70
 - ISPF prototyping 74
 - Data Element Facility 77
 - data structures 83
 - APS-generated report 170
 - naming conventions 19
 - Database Access Summary screen 132
 - Database Call Tailoring screen 134
 - database calls 31
 - IDMS DB 32
 - IMS DB 36
 - SQL 41
 - VSAM batch 50
 - VSAM online 53
 - Database Qualification screen 136
 - Database Record Selection screen 137
 - date field edits 213
 - DB target 18
 - DB-BIND call
 - IDMS DB 32
 - DB-CLOSE call
 - IDMS DB 32
 - SQL 42
 - VSAM batch 50
 - DB-COMMIT call
 - SQL 42
 - DB-DECLARE call
 - SQL 41, 48, 49
 - DB-ERASE call
 - IDMS DB 32
 - IMS DB 36
 - SQL 44
 - VSAM batch 50
 - VSAM online 53
 - DB-FETCH call
 - SQL 44
 - DB-FREE call
 - VSAM online 53
 - DB-GET call
 - IDMS DB 32
 - DB-MODIFY call
 - IDMS DB 33
 - IMS DB 36
 - SQL 44
 - VSAM batch 51
 - VSAM online 53
 - DB-OBTAIN call
 - IDMS DB 33
 - IMS DB 37
 - SQL 41, 44, 48
 - VSAM batch 51
 - VSAM online 53
 - DB-OPEN call
 - IDMS DB 34
 - SQL 45
 - VSAM batch 51

- DB-PROCESS call
 - IDMS DB 34
 - IMS DB 38
 - SQL 42, 45, 49, 50
 - VSAM batch 51
 - VSAM online 54
- DB-ROLLBACK call
 - SQL 47
- DB-STORE call
 - IDMS DB 34
 - IMS DB 38
 - SQL 47
 - VSAM batch 52
 - VSAM online 54
- DC target 18
- DDI syntax
 - IMS DB 105
 - VSAM batch 106
 - VSAM online 106
- DECL keyword 114
- DECLARE call
 - SQL 41, 48, 49
- DECLARE statement 26
- DEFINE statement 27
- DEFINED function 26
- DEFVAL function 26
- deleting
 - application components 13
 - field edits, screen fields 215
 - screen fields 221
- design options
 - Screen Painter 206
- development steps, application 8
- DLG- calls 70
- DPAR keyword 114
- DS keyword 114

E

- edit masks in data structures 85
- editing APS components 13

- edits, field
 - time 213, 218
- END statement 27
- ENDEVOR interface 87
- END-PROCESS flag
 - VSAM batch 52
 - VSAM online 55
- ENTRY statement 195
- EPILOGUE function 26
- ERASE call
 - IDMS DB 32
 - IMS DB 36
 - SQL 44
 - VSAM batch 50
 - VSAM online 53
- error handling
 - CICS 61
 - Customization Facility 27
 - IDMS DB 34
 - IMS DB 39
 - IMS DC 65
 - screen fields 215
 - SQL 47
 - VSAM batch 52
 - VSAM online 54
- ESCAPE statement
 - Customization Facility 27
 - S-COBOL 195
- EVALUATE statement 123, 195
- EXIT PROGRAM statement 196
- Express Parameters screen 138

F

- FALSE statement 199
- FD keyword 115
- FETCH call
 - SQL 44
- field attributes, screen fields 207
- field edits
 - control file 23
 - screens 208

Field Mapping screen 139

fields

VSAM batch 52

VSAM online 55

flags

CICS 62

IMS DC 67

ISPF Dialog 72

S-COBOL 198

VSAM batch 52

VSAM online 55

FREE call

VSAM online 53

FRFM keyword 115

FULL function 26

G

generating

CICS screen parameters 223

IDMS DB options 101

IMS DC screen parameters 224, 226

ISPF Dialog screen parameters 228

ISPF prototyping screen parameters 224

Options screen 95

parameters for screens 222

precompiler options 96

single-platform applications 93

SQL options 99

when to regenerate 102

GET call

IDMS DB 32

H

HALF function 26

help

creating for end users 229

online facility for APS 15

I

IDCS keyword 115

IDM- calls 35

IDMS DB

application generation options 101

database calls 32

IDMS Options screen 101

IDSS keyword 115

IF/ELSE-IF/ELSE statement

Customization Facility 27

S-COBOL 196

IM- calls 40, 66, 69

importing databases and files 105

APS-generated report 170

IMS DB

control file 23

database calls 36

DDI syntax 105

IMS DC

control file 23

data communications calls 65

screen generation parameters 224, 226

INCLUDE statement 28

INDENT function 26

INDEX function 26

INVALID KEY statement 195

IO keyword 116

ISPF Dialog

control file 23

data communications calls 70

screen generation parameters 228

ISPF prototyping

data communications calls 74

screen generation parameters 224

J

Job Control Cards screen 100

Joins 48

K

keywords, Program Painter and Specification Editor 109

L

LENGTH function 26

limits, APS 121

LINK call

CICS 62

IMS DC 67

ISPF Dialog 72

ISPF prototyping 75

LINKAGE call

CICS 64

IMS DC 69

ISPF Dialog 74

list boxes

program functions for, specifying 161

LK keyword 116

LOOKUP statement 28

M

Main Menu screen 12

Menu Item Functions screen 141

MOCK keyword 116

MODIFY call

IDMS DB 33

IMS DB 36

SQL 44

VSAM batch 51

VSAM online 53

MSG-SW call

IMS DC 67

ISPF prototyping 75

N

naming conventions 19

navigating

APS Painters 7, 17, 201

Online Express 125

Screen Painter 205

NTRY call

CICS 62

IMS DC 67

ISPF Dialog 72

ISPF prototyping 75

NULL call

CICS 64

IMS DC 69

ISPF Dialog 74

numeric field edits

screen fields 216

NUMERIC function 26

O

OBTAIN call

IDMS DB 33

IMS DB 37

SQL 41, 44, 48

VSAM batch 51

VSAM online 53

OCCURS clause 85

Online Express 125

limits 122

online help facility for APS 15

OPEN call

IDMS DB 34

SQL 45

VSAM batch 51

OPT keyword 116

Order by Columns screen 144

P

- Painter Menu screen 13
- PARA keyword 117
- parameters
 - application generation 95
 - CICS screen generation 223
 - IDMS DB generation 101
 - IMS DC screen generation 224, 226
 - ISPF Dialog screen generation 228
 - ISPF prototyping screen generation 224
 - precompiler 96
 - screen generation 222
 - SQL generation 99
- PARSE function 26
- PERFORM call
 - CICS 64
 - IMS DC 70
 - ISPF Dialog 74
 - ISPF prototyping 76
- PERFORM statement 197
- PF Key Functions screen 144
- PF key values
 - CICS 62
 - IMS DC 67
 - ISPF Dialog 73
- picture formats 85
- Precompiler Options screen 96
- printing APS reports 13
- PROC keyword 117
- PROCESS call
 - IDMS DB 34
 - IMS DB 38
 - SQL 42, 45, 49, 50
 - VSAM batch 51
 - VSAM online 54
- Program Definition screen 146
- program naming conventions 19
- Program Painter
 - APS-generated report 173
- Program Painter keywords 109
- PROGRAM-ID function 26

- Project Goup Environment screen 96
- PSB-NAME function 26

R

- ranges of values
 - screen fields 218
- REC keyword 117
- RED keyword 117
- REDEFINES clause 86
- regenerating applications 102
- REM keyword 117
- RENAMES clause 84
- renaming application components 14
- REPEAT statement
 - S-COBOL 197
- repeated record blocks 220
- Report Writer
 - APS-generated report 174
 - limits 123
- reports
 - APS-generated 169
 - ENDEVOR 90
 - naming conventions 19
- reserved words 189
- RESET-OBTAIN flag
 - VSAM batch 52
 - VSAM online 56
- RESET-PFKEY call
 - CICS 63
 - IMS DC 68
 - ISPF Dialog 73
 - ISPF prototyping 75
- ROLLBACK call
 - SQL 47

S

- Savekey Definition screen 148
- SC-CLEAR call
 - CICS 61
 - IMS DC 65
 - ISPF Dialog 70
- Scenario Painter 201
 - APS-generated report 174
 - limits 122
- SC-NULL call
 - CICS 64
 - IMS DC 69
 - ISPF Dialog 74
- S-COBOL 193
 - limits 123
- Screen Painter 206
 - APS-generated reports 172, 175
 - limits 122
- screen size field 14
- screens
 - generation parameters 222
 - naming conventions 19
 - size field 18
- screens, APS
 - Alternate Functions 127
 - Application Painter 17
 - Checkin 87
 - Checkout 89
 - Column Selection 129
 - Control Points 130, 158
 - Correlation Names 131
 - Data Element Facility 77
 - Database Access Summary 132
 - Database Call Tailoring 134
 - Database Qualification 136
 - Database Record Selection 137
 - Express Parameters 138
 - Field Mapping 139
 - Generator Options 95
 - IDMS Options 101
 - Job Control Cards 100
 - Main Menu 12
 - Menu Item Functions 141
 - Online Express 125
 - Online Express Information Summary 142
 - Online Express Menu 143
 - Order by Columns 144
 - Painter Menu 13
 - PF Key Functions 144
 - Precompiler Options 96
 - Program Definition 146
 - Project Goup Environment 96
 - Savekey Definition 148
 - Screen Generation Parameters 222
 - Special Key Definition 148
 - Specification Editor 150
 - SQL Bind Options 99
 - SQL Command Review 151
 - SQL Functions 151
 - SQL Having Clause Specification 152
 - SQL Prototype 152
 - SQL Qualification Specification 153
 - Subselect Specification 155
 - Union Columns Cross Reference 157
 - Union Summary Menu 156
 - User Controlled Database Calls 158
- SCRNLIST call
 - CICS 63
 - IMS DC 68
 - ISPF Dialog 73
 - ISPF prototyping 75
- SD keyword 118
- SEARCH statement 198
- SEND call
 - CICS 63
 - IMS DC 68
 - ISPF Dialog 73
 - ISPF prototyping 75
- SET Customization Facility statements 29
- Special Key Definition screen 148
- special registers, SQL 49
- Specification Editor
 - keywords 109
- Specification Editor screen 150
- SPNM keyword 118

SQL

- application generation options 99
- Command Review screen 151
- control file 23
- database calls 41
- Functions screen 151
- Having Clause Specification screen 152
- keyword 118
- Prototype screen 152
- Qualification Specification screen 153
- SQL Bind Options screen 99
- STOP RUN statement 199
- STORE call
 - IDMS DB 34
 - IMS DB 38
 - SQL 47
 - VSAM batch 52
 - VSAM online 54
- STUB keyword 118
- Subselect Specification screen 155
- SUBSTR function 26
- SUPRA statement 199
- SY keywords 118

T

targets

- DB for application 18
- DC for application 18
- TERM call
 - CICS 63
 - IMS DC 69
 - ISPF Dialog 74
 - ISPF prototyping 76
- time field edits 218
- TP-ATTR call
 - CICS 58
 - IMS DC 65
 - ISPF Dialog 70
 - ISPF prototyping 74
- TP-BACKOUT call
 - CICS 64

- IMS DC 69
- TP-CLEAR call
 - CICS 61
 - IMS DC 65
 - ISPF Dialog 70
- TP-CLEAR-ATTRS call
 - IMS DC 65
- TP-CLEAR-ATTR call
 - CICS 61
- TP-ENTRY call
 - CICS 62
 - IMS DC 67
 - ISPF Dialog 72
 - ISPF prototyping 75
- TP-LINK call
 - CICS 62
 - IMS DC 67
 - ISPF Dialog 72
 - ISPF prototyping 75
- TP-LINKAGE call
 - CICS 64
 - IMS DC 69
 - ISPF Dialog 74
- TP-MSG-SW call
 - IMS DC 67
 - ISPF prototyping 75
- TP-NULL call
 - CICS 64
 - IMS DC 69
 - ISPF Dialog 74
- TP-PERFORM call
 - CICS 64
 - IMS DC 70
 - ISPF Dialog 74
 - ISPF prototyping 76
- TP-RESET-PFKEY call
 - CICS 63
 - IMS DC 68
 - ISPF Dialog 73
 - ISPF prototyping 75
- TP-SCRNLIST call
 - CICS 63
 - IMS DC 68
 - ISPF Dialog 73

- ISPF prototyping 75
- TP-SEND call
 - CICS 63
 - IMS DC 68
 - ISPF Dialog 73
 - ISPF prototyping 75
- TP-TERM call
 - CICS 63
 - IMS DC 69
 - ISPF Dialog 74
 - ISPF prototyping 76
- TP-XCTL call
 - CICS 64
 - ISPF prototyping 76
- TRACE-FLAG 198
- TRUE statement 199

U

- unions 49
 - Columns Cross Reference screen 157
 - Summary Menu screen 156
- UNTIL statement
 - Customization Facility 30
 - S-COBOL 200
- User Controlled Database Calls screen 158
- User Help facility 229
 - control file 23
- user-defined field edits
 - screen fields 210
- USERNAME statement 200

V

- Validate command 119
- value ranges
 - screen fields 218
- values and conversions
 - screen fields 213

- VSAM batch
 - control file 23
 - database calls 50
 - DDI syntax 106
- VSAM online
 - control file 23
 - database calls 53
 - DDI syntax 106

W

- WHILE statement
 - Customization Facility 30
 - S-COBOL 200
- WS keyword 119

X

- XCTL call
 - CICS 64
 - ISPF prototyping 76

